

日本国特許庁
PATENT OFFICE
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日
Date of Application: 2000年 5月 1日

出願番号
Application Number: 特願2000-132694

出願人
Applicant(s): 株式会社東芝

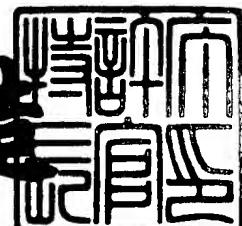


CERTIFIED COPY OF
PRIORITY DOCUMENT

2001年 3月 9日

特許庁長官
Commissioner,
Patent Office

及川耕三



【書類名】 特許願
【整理番号】 A000002470
【提出日】 平成12年 5月 1日
【あて先】 特許庁長官 殿
【国際特許分類】 G06F 3/00
【発明の名称】 画面制御プログラム及び動的表示情報取得プログラム並びに画面表示トランザクションプログラムを記録したコンピュータ読み取り可能な記録媒体
【請求項の数】 6
【発明者】
【住所又は居所】 東京都府中市東芝町1番地 株式会社東芝府中工場内
【氏名】 今村 大輔
【特許出願人】
【識別番号】 000003078
【氏名又は名称】 株式会社 東芝
【代理人】
【識別番号】 100058479
【弁理士】
【氏名又は名称】 鈴江 武彦
【電話番号】 03-3502-3181
【選任した代理人】
【識別番号】 100084618
【弁理士】
【氏名又は名称】 村松 貞男
【選任した代理人】
【識別番号】 100068814
【弁理士】
【氏名又は名称】 坪井 淳

【選任した代理人】

【識別番号】 100092196

【弁理士】

【氏名又は名称】 橋本 良郎

【選任した代理人】

【識別番号】 100091351

【弁理士】

【氏名又は名称】 河野 哲

【選任した代理人】

【識別番号】 100088683

【弁理士】

【氏名又は名称】 中村 誠

【選任した代理人】

【識別番号】 100070437

【弁理士】

【氏名又は名称】 河井 将次

【手数料の表示】

【予納台帳番号】 011567

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【ブルーフの要否】 要

【書類名】 明細書

【発明の名称】 画面制御プログラム及び動的表示情報取得プログラム並びに画面表示トランザクションプログラムを記録したコンピュータ読み取り可能な記録媒体

【特許請求の範囲】

【請求項1】 コンピュータに、

画面に表示される場合に動的に変化する表示内容を得るために予め作成されている少なくとも一つの動的表示処理のうちのいずれかを指定する動的表示処理識別情報と、この動的表示処理識別情報で指定されている動的表示処理に提供される表示属性情報を取得させる動的表示情報取得機能と、

前記動的表示情報取得機能によって取得された動的表示処理識別情報で指定されている動的表示処理に対して該当する前記表示属性情報を提供させる属性情報提供機能と

を実現させるための画面制御プログラムを記録したコンピュータ読み取り可能な記録媒体。

【請求項2】 請求項1記載の画面制御プログラムを記録したコンピュータ読み取り可能な記録媒体であって、

前記属性情報提供機能は、前記動的表示情報取得機能によって取得された動的表示処理識別情報で指定されている動的表示処理を組み込むことを特徴とする画面制御プログラムを記録したコンピュータ読み取り可能な記録媒体。

【請求項3】 請求項1又は請求項2記載の画面制御プログラムを記録したコンピュータ読み取り可能な記録媒体であって、

前記少なくとも一つの動的表示処理のいずれかは、表示属性情報の内容に基づいて検索を行う処理であることを特徴とする画面制御プログラムを記録したコンピュータ読み取り可能な記録媒体。

【請求項4】 コンピュータに、

画面に表示される場合に動的に変化する表示内容を得るために予め作成されている少なくとも一つの動的表示処理のうちのいずれかを指定する動的表示処理識

別情報と、この動的表示処理識別情報で指定されている動的表示処理に提供される表示属性情報を取得させる取得機能と、

前記動的表示処理に対して該当する表示属性情報を提供する機能に対して、前記取得機能によって取得された動的表示処理識別情報と表示属性情報を提供する提供機能と

を実現させるための動的表示情報取得プログラムを記録したコンピュータ読み取り可能な記録媒体。

【請求項5】 請求項4記載の動的表示情報取得プログラムを記録したコンピュータ読み取り可能な記録媒体であって、

前記動的表示処理識別情報と前記表示属性情報とは、所定のフォーマットで記述されており、

コンピュータに、

前記取得機能と前記提供機能との間において、前記所定のフォーマットで記述されている動的表示処理識別情報と表示属性情報を、前記動的表示処理に対して該当する表示属性情報を提供する機能で取り扱い可能なフォーマットに変換するフォーマット変換機能を付加した動的表示情報取得プログラムを記録したコンピュータ読み取り可能な記録媒体。

【請求項6】 コンピュータに、

表示画面に対する入力内容が正常な時点で、この正常な入力内容を画面表示トランザクション情報として記憶させる記憶機能と、

再表示指示を受け付けた場合に、前記記憶機能によって記憶された画面表示トランザクション情報を参照させ、任意の時点での正常な入力内容を前記表示画面に対して再現させる再現機能と

を実現させるための画面表示トランザクションプログラムを記録したコンピュータ読み取り可能な記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、ディスプレイに表示され、ユーザが対話的に入力する画面を提供す

るための画面制御プログラム、及び動的表示情報取得プログラム、並びに画面表示トランザクションプログラムを記録したコンピュータ読み取り可能な記録媒体に関する。

【0002】

【従来の技術】

ユーザが対話的にデータを入力するための画面プログラムを作成する場合において、従来はプログラム言語で画面プログラムを記述して作成する方法が利用されている。

【0003】

また、他の画面プログラムの作成方法として、開発ツールを用いて画面イメージを作画的に編集し画面に関するプログラムソースを生成する方法や、ファイルなどに画面のレイアウト情報を別途定義して実行時にプログラムがこの画面のレイアウト情報をロードして表示する方法も利用されている。

【0004】

【発明が解決しようとする課題】

上記の3種類の画面プログラム作成方法のうち、プログラム言語により画面プログラムを記述する方法を利用すると、プログラムの自由度は高くなるが記述量が多くなる。また、プログラムを変更する場合にプログラムの編集及びコンパイル等の作業が必要であり、開発効率の向上が困難である。

【0005】

また、開発ツールを用いて画面プログラムを作成する方法を利用した場合、画面のレイアウト作成は容易であるが、画面表示ために必要な処理に関する部分をプログラム言語で記述する必要があり、画面プログラムの作成が困難な場合がある。また、この方法ではレイアウトの変更が必要な場合にコンパイルする必要がある点も上記の場合と同様である。

【0006】

画面のレイアウト情報を定義して実行時にロードする方法を利用した場合には、画面のレイアウトを変更する場合であってもプログラムソースの編集とコンパイルが不要であるためプログラムの開発効率は高い。しかしながら、この方法で

は、プログラム外部に定義できるレイアウト情報の情報量に限りがあり、自由度と拡張性が高くない。例えば、画面を構成する部品の大きさ、位置、色の種類、画面上に配置される文字の指定などに関してはレイアウト情報として定義可能であるが、画面表示のために必要な処理に関する部分はプログラム内部で定義する必要がありプログラム外部で定義されない。

【0007】

すなわち、上記3種類の画面プログラム作成方法を利用する場合には、画面表示のために必要な処理に関する部分をプログラム言語によりプログラミングする必要があり、また変更時にコンパイルする必要があるため、開発労力がかかるという問題がある。

【0008】

加えて、業務アプリケーションで問題になるのは、障害発生時のアプリケーションの状態管理である。業務処理の障害に応じてデータベースの内容を復帰させることは通常行われているが、画面についても同様に復帰が望まれる場合がある。業務アプリケーションが障害から復帰した場合に、障害発生前において入力済み内容を、ユーザが画面に対して再入力するのは効率的ではない。

【0009】

しかしながら、画面の表示状態を適切な状態に復帰させるプログラム開発には多くの労力がかかる。

【0010】

本発明は、以上のような実情に鑑みてなされたもので、拡張性の高い画面プログラムを効率的かつ容易に開発し、画面プログラムの耐障害性を向上させるための画面制御プログラム、及び動的表示情報取得プログラム、並びに画面表示トランザクションプログラムを記録したコンピュータ読み取り可能な記録媒体に関する。

【0011】

【課題を解決するための手段】

本発明の骨子は、画面に表示される場合に動的に変化する表示内容を得るために少なくとも一つの動的表示処理が予め作成されており、この動的表示処理の指

定及びこの動的表示処理に提供する属性の指定によって画面を構成可能な点にある。

【0012】

本発明を実現するにあたって講じた具体的手段について以下に説明する。

【0013】

第1の発明は、コンピュータに、画面に表示される場合に動的に変化する表示内容を得るために予め作成されている少なくとも一つの動的表示処理のうちのいずれかを指定する動的表示処理識別情報と、この動的表示処理識別情報で指定されている動的表示処理に提供される表示属性情報とを取得させる動的表示情報取得機能と、動的表示情報取得機能によって取得された動的表示処理識別情報で指定されている動的表示処理に対して該当する表示属性情報を提供させる属性情報提供機能とを実現させるための画面制御プログラムを記録したコンピュータ読み取り可能な記録媒体である。

【0014】

この第1の発明における画面制御プログラムを利用すると、ユーザは、表示する時点で動的に変化する内容についても、画面プログラム開発時あるいは変更時に動的表示処理の指定と表示属性情報の指定を行うのみで画面上に表示させることができる。

【0015】

したがって、画面表示のために必要な処理に関する部分をプログラム言語によりプログラミングする必要がなく、容易に画面プログラムを作成でき、開発時あるいは変更時にコンパイルする必要もない。

【0016】

これにより、ユーザは拡張性の高い画面プログラムを効率的に開発でき、画面プログラムを容易に変更できる。

【0017】

第2の発明は、第1の発明と同様の画面制御プログラムを記録したコンピュータ読み取り可能な記録媒体であるが、属性情報提供機能は、動的表示情報取得機能によって取得された動的表示処理識別情報で指定されている動的表示処理を組

み込むという特徴を持つ。

【0018】

すなわち、この第2の発明においては、予め作成済みの動的表示処理にうち、指定されたものが属性情報提供機能に対してプラグインされ、表示属性情報に基づく利用が可能となる。

【0019】

これにより、ユーザは画面プログラム開発時あるいは変更時にプログラミングする必要がない。すなわち、ユーザは、単に動的表示処理と表示属性情報を指定するのみで画面プログラムにおける処理に関する部分についても開発あるいは変更可能となる。

【0020】

第3の発明は、第1又は第2の発明と同様の画面制御プログラムを記録したコンピュータ読み取り可能な記録媒体であり、少なくとも一つの動的表示処理のいずれかは、表示属性情報の内容に基づいて検索を行う処理であるという特徴を持つ。

【0021】

これにより、あるキーにしたがって検索されたデータを画面に表示させる場合であっても、ユーザはこの検索を行う動的表示処理を指定し、そのキーを表示属性情報として指定するのみで、検索結果を画面上に表示できる。したがって、上記と同様の効果を得ることができる。

【0022】

なお、動的表示処理の他の例としては、入力内容をチェックしてエラーの場合にその旨を返す入力チェック処理、新規の画面をポップアップするポップアップ処理などがある。

【0023】

第4の発明は、コンピュータに、画面に表示される場合に動的に変化する表示内容を得るために予め作成されている少なくとも一つの動的表示処理のうちのいずれかを指定する動的表示処理識別情報と、この動的表示処理識別情報で指定されている動的表示処理に提供される表示属性情報とを取得させる取得機能と、動

的表示処理に対して該当する表示属性情報を提供する機能に対して、取得機能によって取得された動的表示処理識別情報と表示属性情報を提供させる提供機能とを実現させるための動的表示情報取得プログラムを記録したコンピュータ読み取り可能な記録媒体である。

【0024】

この第4の発明における動的表示情報取得プログラムは、先の第1の発明における画面制御プログラムの動的表示情報取得機能に対応するプログラムである。

【0025】

第5の発明は、第4の発明と同様の動的表示情報取得プログラムを記録したコンピュータ読み取り可能な記録媒体であるが、動的表示処理識別情報と表示属性情報とは所定のフォーマットで記述されており、コンピュータに、取得機能と提供機能との間において、所定のフォーマットで記述されている動的表示処理識別情報と表示属性情報を、動的表示処理に対して該当する表示属性情報を提供する機能で取り扱い可能なフォーマットに変換させるフォーマット変換機能を付加した特徴を持つ。

【0026】

これにより、例えばXML(Extensible Markup Language)ファイル、HTML(HyperText Markup Language)ファイル、CSVファイル等のような所定の形式で記述されたファイルとして動的表示処理識別情報と表示属性情報を受け付け、その内容を後段の機能で取り扱い可能に変換し、提供できる。

【0027】

第6の発明は、コンピュータに、表示画面に対する入力内容が正常な時点で、この正常な入力内容を画面表示トランザクション情報として記憶させる記憶機能と、再表示指示を受け付けた場合に、記憶機能によって記憶された画面表示トランザクション情報を参照させ、任意の時点での正常な入力内容を表示画面に対して再現させる再現機能とを実現させるための画面表示トランザクションプログラムを記録したコンピュータ読み取り可能な記録媒体である。

【0028】

この第6の発明における画面表示トランザクションプログラムは、上記の動的

表示処理の一つとして動作可能である。この画面表示トランザクションプログラムを利用すると、画面への入力作業中に障害が発生しても、障害から復帰した場合に障害発生前の任意の時点での正常な入力内容が画面上に再表示される。

【0029】

したがって、ユーザの操作が効率化され、画面プログラムの耐障害性を容易に向上させることができる。

【0030】

上記のようなプログラムを記録した記録媒体を用いることによって、上述した機能を有していない計算機システム、サーバやクライアント等の計算機に対して簡単に上述した機能を付加することができる。

【0031】

【発明の実施の形態】

以下、図面を参照しながら本発明の実施の形態について説明する。

【0032】

(第1の実施の形態)

本実施の形態においては、レイアウト情報に加えて、画面表示のために必要な処理に関する部分を指定する動的表示情報を取得し、アプリケーション・プログラム（以下、「アプリケーション」という）の画面を表示する画面制御プログラムについて説明する。

【0033】

図1は、本実施の形態に係る画面制御プログラムが計算機上で実現する機能を示すブロック図である。

【0034】

本実施の形態に係る画面制御プログラム1は、主にレイアウト情報取得機能2、動的表示情報取得機能3、制御機能4、検索処理インターフェース5a、入力チェック処理インターフェース5b、ポップアップ処理インターフェース5c、データ入出力機能6、画面表示トランザクション機能7を計算機や計算機システム上で提供する。

【0035】

また、この画面制御プログラム1は、例えば検索処理や、入力チェック処理、ポップアップ処理などのように、画面に表示される場合に動的に変化する表示内容を得るために予め作成されている動的表示処理（以下、「コンポーネント」という）8a～8cを利用する。コンポーネント8a～8cは既にコンパイルされており、そのままの状態で実行可能なプログラムである。

【0036】

ユーザ（画面表示プログラム作成者）は、アプリケーション11の画面についてのレイアウト情報9と動的表示情報10とを入力する。

【0037】

レイアウト情報9には、先に述べたように、例えば画面を構成するための部品の配置位置、大きさ、色などのような、画面を構成する場合に固定的に決定される静的情報が主に記述されている。

【0038】

動的表示情報10には、ユーザが利用したいコンポーネントを指定した動的表示処理識別情報や、このコンポーネントの処理内容を決定するために必要な表示属性情報が記述されている。ここでは、例として検索処理、入力チェック処理、ポップアップ処理を指定する旨の動的表示処理識別情報が記述されており、また検索処理において図示しないテーブルT₁の内容が検索対象となる旨の表示属性情報が記述されている。

【0039】

レイアウト情報取得機能2は、レイアウト情報9に定義された内容を解析し、解析結果を制御機能4に伝える。

【0040】

動的表示情報取得機能3は、動的表示情報10に定義された内容を解析し、解析結果を制御機能4に伝える。この動的表示情報取得機能3は、主に取得機能3aと、変換機能3bと、提供機能3cとから構成されている。

【0041】

所定のフォーマットで記述されている動的表示情報10が取得機能3aによって取り込まれると、その記述内容は変換機能3bによって制御機能4で取り扱い

可能なフォーマットに変換され、提供機能3cによって制御機能4に提供される

【0042】

制御機能4の備える属性情報提供機能4aは、動的表示情報取得機能3から提供された動的表示処理識別情報によって指定されているコンポーネントをプラグインするための制御を行い、プラグインされたコンポーネントに対して該当する表示属性情報を提供し、結果を取得する。

【0043】

また、制御機能4は、画面上への入出力用のフィールド（画面上で入出力をを行うための部品）の表示や動作の制御を行う。

【0044】

検索処理インターフェース5aは、検索処理用のコンポーネント8aをプラグインし、動的表示情報10に定義された検索用の表示属性情報に基づいてファイルやデータベースのテーブルを検索し、検索結果を制御機能4に返却する。

【0045】

入力チェック処理インターフェース5bは、入力チェック処理用のコンポーネント8bをプラグインし、レイアウト情報9もしくは動的表示情報10に定義された入力チェック用の表示属性情報に基づいてフィールドに入力された値をチェックし、その結果を制御機能4に返却する。

【0046】

ポップアップ処理インターフェース5cは、ポップアップ処理用のコンポーネント8cをプラグインし、動的表示情報10に定義されたポップアップ処理用の表示属性情報に基づいてポップアップウィンドウを表示し、ポップアップウィンドウにて決定されたデータを制御機能4に反映する。

【0047】

データ入出力機能6は、画面上のフィールドにおいて入力されたデータを取得し、またアプリケーション11から提供されたデータを画面上のフィールドにセットする。

【0048】

画面表示トランザクション処理機能7は、入出力の状態が正常な場合にその入出力状態を示すオブジェクトを保存する。また、画面に対するトランザクションの開始を指示された場合に、保存されているオブジェクトを用いてトランザクションの失敗による画面表示の復帰を行う。

【0049】

以下に、上記構成により実現される動作を具体的に説明する。

【0050】

図2は、レイアウト情報9の具体例を示す図である。このレイアウト情報9の例では、XMLを用いて記述されている場合を示しているが、その他にもCSV形式、特定の文字をデリミタとする形式、固定長ファイル形式が利用できる。

【0051】

図3は、本実施の形態に係る画面制御プログラム1がレイアウト情報9を読み込んだ場合の処理手順を例示する図である。

【0052】

まず、アプリケーション11が、レイアウト情報取得機能2に対してレイアウト情報9をセットする(301)。レイアウト情報取得機能2は、先の図2に示すレイアウト情報9の2行目～21行目に示す<field>～</field>を1フィールド分の定義として展開する(302)。XML文書の展開にはXMLパーザといった既存の技術を用いることができる。

【0053】

展開されたフィールド毎のレイアウト情報は、レイアウト情報取得機能2から制御機能4に対してセットされる(303)。

【0054】

制御機能4は、セットされたレイアウト情報に従ってフィールドを作成する(304)。以下、処理302～304が<field>タグの数だけ繰り返される(305)。

【0055】

図4は、<field>タグの数だけ繰り返される各フィールドの作成処理を例示するフローチャートである。

【0056】

制御機能4は、先の図2に示すレイアウト情報9の2行目の〈field〉タグ内に指定されているIDに関連付けたフィールドオブジェクトを作成する(401)。

【0057】

次に、先の図2に示すレイアウト情報9の3行目～11行目に定義されている位置属性などのフィールドの表示に関する情報を作成されたフィールドオブジェクトに対してセットする(402)。

【0058】

例えば、制御機能4は、先の図2に示すレイアウト情報9の7行目の〈label〉タグに囲まれた属性により、フィールドのラベルをフィールドオブジェクトにセットする。

【0059】

次に、制御機能4は、レイアウト情報9に入力チェック属性が定義されているか否かを判断する(403)。先の図2に示すレイアウト情報9については、13行目の〈validation〉タグの有無により判断できる。

【0060】

ここで、入力チェック属性が指定されている場合には、先の図2に示すレイアウト情報9の15行目の〈component〉タグで指定された入力チェック用のコンポーネントを作成するか、あるいはコンポーネントのインスタンスを取得する(404)。Java言語の場合、クラス名からオブジェクトの作成が可能であり、この処理404に用いることができる。

【0061】

次に、制御機能4は、入力チェック用コンポーネントをフィールドに登録する(405)。

【0062】

次に、制御機能4は、先の図2に示すレイアウト情報9の14行目に示す入力チェックのトリガーとなるキーと、16行目～19行目に示すエラー時の動作とをフィールドに登録する(406)。

【0063】

そして、制御機能4は、フィールドを先の図2に示すレイアウト情報9の12行目に指定するように初期状態にして完了する(407)。この例では、初期状態は「UNEDITABLE」となっている。

【0064】

図5は、上記のようなレイアウト情報9に基づく制御機能4の処理によって構成される画面を示す図である。

【0065】

レイアウト情報9に基づいて作成された画面12は、ラベル12a～12cや初期状態のフィールド12d～12fなどのような不变の画面要素を表示している。

【0066】

例えば、先の図2に示すレイアウト情報9の7行目の<label>タグに囲まれた属性はフィールド12dのラベル12aを示しているが、表示されると「数量」となる。

【0067】

また、先の図2に示すレイアウト情報9の12行目において、初期状態は「UNEDITABLE」とする旨が指定されているため、フィールド12dの初期状態は編集不可能状態となる。

【0068】

図6は、画面制御プログラム1がレイアウト情報9を読み込んだ場合の処理手順の変形例を示す図である。

【0069】

この変形例では、アプリケーション11が、レイアウト情報取得機能2に対してレイアウト情報9をセットし(601)、<field>～</field>までが1フィールド分の定義として展開する(602)。

【0070】

次に、レイアウト情報取得機能2が1フィールド分のレイアウト情報の格納されたオブジェクトを作成し(603、604)、そのオブジェクトを制御機能4

にセットし（605）、フィールドが作成されて属性がセットされる（606）。

【0071】

その後、処理602から606までの動作がフィールドの数だけ繰り返される（607）。

【0072】

図7は、レイアウト情報9において定義された入力チェック属性に従って画面制御プログラム1が実行する入力チェック処理の処理手順を例示する図である。

【0073】

ユーザは、フィールドに値を入力した後、入力チェックのトリガーとなる操作を行う（701）。先の図2に示すレイアウト情報9の14行目に〈trigger〉タグで囲まれた属性として「VK_ENTER」が指定されているが、これはエンターキーの入力が入力チェックのトリガーになることを示している。

【0074】

このトリガーキーの入力によって、制御機能4は、フィールドに入力された値を入力チェック処理インターフェース5bに対して渡し、入力のチェックを指示する（702）。

【0075】

入力チェック処理インターフェース5bは、値が入力されたフィールドに登録された入力チェック処理用のコンポーネント8bに対して入力チェックを指示する（703）。

【0076】

入力チェック処理用のコンポーネント8bは、渡された値を自分の実装により判断し、エラーか否かを返却する（704）。

【0077】

入力チェック処理インターフェース5bは、得られた結果を制御機能4に返却する（705）。

【0078】

制御機能4は、チェック結果がエラーだった場合に先の図2に示す16行目～

19行目に定義された内容に従ってフィールドの属性変更を行う。

【0079】

このような動作は、フィールドを構成するプログラムに予め用意されており、レイアウト情報9が記述されているファイルでは、それらの中から選択的に指定している。この例では、<type>タグに「CHANGE_BACKGROUND_COLOR」と定義されており、また<param>タグで「background」が#FF0000と指定されている。

【0080】

したがって、図8に例示するように、入力チェックの結果がエラーだった場合には、そのフィールド12dの背景色が赤になる。

【0081】

続いて、本実施の形態において特徴的な情報である動的表示情報10について説明する。なお、上記のレイアウト情報9において説明した入力チェック属性は、この動的表示情報10で指定されてもよい。

【0082】

図9は、動的表示情報10の具体例を示す図である。

【0083】

この動的表示情報10の例では、レイアウト情報9と同様にXMLを用いて記述された場合を示しているが、他にもCSV形式、特定の文字をデリミタとする形式、固定長ファイル形式で記述してもよい。

【0084】

図10は、本実施の形態に係る画面制御プログラム1が動的表示情報10を読み込んだ場合の処理手順を例示する図である。

【0085】

まず、アプリケーション11が、動的表示情報取得機能3に対して動的表示情報10をセットする(1001)。動的表示情報取得機能3は、先の図9に示す動的表示情報10の2行目～13行目に示す<field>～</field>を1フィールド分の定義として展開する(1002)。XML文書の展開にはレイアウト情報9の場合と同様にXMLパーザといった既存の技術を用いることができる。

【0086】

展開されたフィールド毎の動的表示情報は、動作表示情報取得機能3から制御機能4に対してセットされる(1003)。

【0087】

制御機能4は、セットされた動的表示情報に従ってフィールドの表示属性をセットする(1004)。以下、処理1002～1004の処理が〈field〉タグの数だけ繰り返される(1005)。この結果、画面上の個々のフィールドについての表示属性が変更され、かつ動作に必要な情報が各フィールドにセットされる。

【0088】

図11は、上記のような動的表示情報10に基づく制御機能4の処理によって構成される画面を示す図である。

【0089】

この画面12においては、先の図9に示す動的表示情報10の3行目に指定するように、「area」というIDのフィールドは〈state〉タグで囲まれた「COMBO」という表示属性になる。この「COMBO」という表示属性は、一覧からの選択を可能としつつテキストフィールドに対して直接の文字入力を可能とする旨を示す動的表示処理識別情報であり、画面要素14aに示すような動的な表示形態をとる。

【0090】

また、先の図9に示す動的表示情報10の15行目に指定する「product」というIDを持つフィールドは「POPUP」という表示属性である。この「POPUP」はテキストフィールドにポップアップウィンドウを表示する時に押下するためのボタンを付属する旨を示す動的表示処理識別情報であり、画面要素14bに示すような動的な表示形態をとる。このフィールドにセットされる表示属性毎の処理については後述する。

【0091】

図12は、画面制御プログラム1が動的表示情報10を読み込んだ場合の処理手順の変形例を示す図である。

【0092】

この変形例では、アプリケーション11が、動的表示情報取得機能3に対して動的表示情報10をセットし(1201)、<field>～</field>までを1フィールド分の定義として展開する(1202)。

【0093】

次に、動的表示情報取得機能3が1フィールド分の動的表示情報の格納されたオブジェクトを作成し(1203、1204)、そのオブジェクトを制御機能4にセットし(1205)、フィールドへの属性がセットされる(1206)。

【0094】

そして、処理1202から1206と同様の動作がフィールドの数だけ繰り返される(1207)。

【0095】

図13は、フィールドにセットされた表示属性が検索だった場合に画面制御プログラム1が実行する処理の手順を例示する図である。

【0096】

また、図14は、フィールドにセットされた表示属性が検索だった場合の画面制御プログラム1の処理を例示するフローチャートである。

【0097】

制御機能4に動的表示情報10がセットされると、制御機能4は各フィールド毎にフィールドの表示属性を変更する(1401)。

【0098】

制御機能4は、フィールドの表示属性が「COMBO」、「PULLDOWN」といった一覧選択を行うものであるか否か(検索についての動的表示処理識別情報か否か)を判断し(1402)、その場合は一覧に格納するデータを検索するのに必要な属性(表示属性情報)が動的表示情報にあるか否かを判断する(1403)。

【0099】

検索に必要な属性がある場合には、検索処理インターフェース5aにテーブル名、検索するカラム名、検索条件といった検索のための属性を渡して検索の指示を出す(1301)。

【0100】

検索処理インターフェース5aは、プラグインされた検索処理用のコンポーネント8aを用いて検索処理を行う（1302）。プラグインされた検索処理用のコンポーネント8aは、渡された検索のための属性を利用し自らの実装に基づいてデータベースやファイルのテーブルを検索する（1303、1404）。

【0101】

検索結果は、検索処理用のコンポーネント8aから検索処理インターフェース5aを経由し（1304）、制御機能4に渡され（1305）、フィールドの一覧データに格納される（1306）。

【0102】

上記の処理の結果、図15に示すように、検索処理用のコンポーネント8aによって動的な表示内容が取得され、画面12のフィールド12eにユーザの操作に基づいて検索された一覧15aが表示可能となる。

【0103】

図16は、フィールドにセットされた表示属性がポップアップだった場合に画面制御プログラム1が実行する処理の手順を例示する図である。

【0104】

制御機能4に動的表示情報がセットされると、制御機能4は、検索処理の場合と同様にフィールドの表示属性を変更し、フィールドの表示属性がポップアップを示すものか否か（ポップアップについての動的表示処理識別情報か否か）を判断する。例えば、先の図9に示す動的表示情報10の15行目に指定する「POPP」がポップアップを示す表示属性である。

【0105】

表示属性がポップアップの場合には、ポップアップ処理インターフェース5cに対してポップアップウィンドウに表示するコンポーネント名、ポップアップウィンドウのタイトルバー部分に表示するタイトル、大きさ、ポップアップウィンドウに付属するボタンのラベル名といった属性（表示属性情報）を渡して、ポップアップ処理8cの登録を指示する（1601）。

【0106】

ポップアップ処理インターフェース5cに渡す属性は、先の図9に示す動的18

行目～22行目に指定するように定義可能である。

【0107】

ポップアップ処理インターフェース5cは、プラグインされたポップアップ用のコンポーネント8cを作成、もしくはインスタンスを取得し(1602、1603)、取得したポップアップ用のコンポーネント8cの表示処理をフィールドへの操作の結果起動される処理として登録する(1604)。これはJava言語を例にとると、ボタン押下のイベントにより起動するモジュールをボタンに登録するという処理で実現可能である。

【0108】

図17は、ユーザによるフィールドへの操作の結果、起動されるポップアップ処理の手順を例示する図である。

【0109】

ユーザが制御機能4により表示されているフィールドに対しポップアップ表示を命ずる操作を行う(1701)。

【0110】

制御機能4は、ポップアップ処理インターフェース5cに対し、ポップアップウィンドウの表示を指示する(1702)。

【0111】

ポップアップ処理インターフェース5cは、ポップアップウィンドウを表示しウインドウ上にプラグインされたポップアップ処理用のコンポーネント8cを表示する(1703)。

【0112】

この状態での画面の具体例を図18に示す。この図18における画面12の要素16aがプラグインされたポップアップ処理用のコンポーネント8cである。

【0113】

ここで、ユーザからの入力操作を待ち、ユーザはポップアップウィンドウに対してデータを入力する(1704)。図18の例では、画面要素16bにおける選択操作がこの入力操作に該当する。

【0114】

ユーザによってポップアップウィンドウに付属している決定ボタンの押下といった確定操作が行われると(1705)、ポップアップ処理インターフェース5cはポップアップウィンドウを閉じる(1706)。

【0115】

ポップアップウィンドウに付属している決定ボタンとは、図18の画面要素16cのように、ポップアップウィンドウ自体への操作のために予め用意されているボタンである。

【0116】

そして、ポップアップ処理インターフェース5cは、プラグインされたポップアップ処理用のコンポーネント8cに対しユーザが入力したデータを取得するための指示を発行する(1707)。

【0117】

その結果、ポップアップ処理用のコンポーネント8cは、入力されたデータのIDとデータの値をポップアップ処理インターフェース5cに返却する(1708)。ここでデータのIDとは、ポップアップ処理インターフェース5cとポップアップ処理用のコンポーネント8cとの間の規約により必要とされるもので、制御機能4の任意のフィールドにつけられたIDと同値である。

【0118】

ポップアップ処理インターフェース5cは、得られたデータを反映するフィールドをデータのIDをキーにして検索し、制御機能4の該当するフィールドにセットする(1709)。

【0119】

この状態での画面の具体例を図19に示す。この図19における画面12のフィールド12cには、ユーザがポップアップウィンドウで選択したデータがセットされ、表示される。

【0120】

図20は、画面表示トランザクション機能7を利用して画面表示を復帰させる処理の手順を例示する図である。

【0121】

アプリケーション11は、画面表示トランザクション機能7に対して画面表示トランザクションの開始を指示する(2001)。

【0122】

画面表示トランザクション機能7は、制御機能4に対し、現在の画面状態の取得を指示する(2002)。

【0123】

制御機能4は、現在の画面状態を示す情報として、セットされているレイアウト情報、セットされている動的表示情報、各フィールドの活性データを格納した画面表示トランザクション情報オブジェクトを作成する(2003、2004)。ここで、活性データとは、ユーザがフィールドに入力したデータやアプリケーションの実行によってフィールドに表示されたデータを指す。

【0124】

制御機能4は、画面表示トランザクション情報オブジェクトを画面表示トランザクション機能7に返却し(2005)、画面表示トランザクション機能7はアプリケーション11に画面表示トランザクション情報オブジェクトを返却する(2006)。

【0125】

この時点での画面表示トランザクション情報オブジェクトは、表1に示すようにレイアウト情報、動的表示情報、活性データを管理している。

【0126】

【表1】

| | |
|---------|------------------|
| レイアウト情報 | [レイアウトオブジェクト] |
| 動的表示情報 | [動的表示情報オブジェクト] |
| 活性データ | [活性データの一覧オブジェクト] |

【0127】

なお、画面表示トランザクション情報オブジェクト自体の返却に代えて、画面

表示トランザクション情報オブジェクトを画面表示トランザクション機能7、もしくは制御機能4によって特定のIDと対応づけて管理し、アプリケーション11にIDのみを返却するとしてもよい。

【0128】

その後、アプリケーション11は、業務処理を続け(2007)、業務処理が失敗し画面状態の復帰が必要になった場合に、画面表示トランザクション機能7に対しトランザクションの復帰を指示する(2008)。この時、アプリケーション11は、画面表示トランザクション情報オブジェクト、もしくは画面表示トランザクション情報オブジェクトを示すIDを画面表示トランザクション機能7に渡して復帰を指示する。

【0129】

画面表示トランザクション機能7は、渡された画面表示トランザクション情報オブジェクト、もしくは画面表示トランザクション情報オブジェクトを示すIDを制御機能4にセットして状態の復帰を指示する(2009)。

【0130】

制御機能4は、画面表示トランザクション情報オブジェクトがセットされると、画面表示トランザクション情報オブジェクトからレイアウト情報、動的表示情報、活性データに基づき、画面の状態をトランザクションが開始された時点の状態を含んだ任意の状態へ復帰する(2010)。または、制御機能4は、画面表示トランザクション情報オブジェクトを示すIDがセットされると、IDから画面表示トランザクション情報オブジェクトを検索し、以下同様の手順で画面の状態を復帰する。

【0131】

以上説明したように、本実施の形態に係る画面制御プログラム1を利用するこことにより、ユーザは、表示を要求する時点毎に動的に変化する表示内容についてもこの動的な表示内容を取得するために予め設けられている動的表示処理を指定し、またこの動的な表示内容を得るための属性を指定するのみで、画面上に表示させることができる。

【0132】

したがって、表示内容が変化する画面プログラムを開発・変更する場合であってもプログラミングする必要がなく、コンパイルする必要もない。

【0133】

ゆえに、画面プログラムを容易かつ効率的に開発することができ、開発労力を軽減できる。

【0134】

また、本実施の形態に係る画面制御プログラム1を利用することにより、アプリケーション11が障害から復帰した場合に画面に障害発生前の正常な入力状態を再現させることができる。

【0135】

したがって、ユーザは画面に再入力する必要がなく、入力操作を効率化させることができ。また、ユーザはこのような入力操作を効率化させる画面プログラムを容易に開発できる。

【0136】

なお、本実施の形態においては、画面制御プログラム1が画面表示トランザクション機能7を備えている場合について説明しているが、これに限定されるものではなく、画面表示トランザクション機能を実現するためのコンポーネントをプラグインする形態を採用してもよい。

【0137】

本実施の形態においてレイアウト情報取得機能2及び動的表示情報取得機能3は、画面を表示する際にそれぞれレイアウト情報9及び動的表示情報10を取得し、変換し、変換結果を制御機能4に提供する。しかしながら、これに代えて、レイアウト情報取得機能2及び動的表示情報取得機能3は、それぞれレイアウト情報9及び動的表示情報10を予め取得、変換しておき、画面を表示する際に制御機能4がこの変換済みの情報を読み込むとしてもよい。

【0138】

(第2の実施の形態)

本実施の形態においては、レイアウト情報のセットを複数回に分けて行う画面制御プログラムについて説明する。

【0139】

図21は、本実施の形態に係る画面制御プログラムが計算機上で実現する機能を示すブロック図であり、図1と同一の部分については同一の符号を付してその説明を省略し、ここでは異なる部分についてのみ詳しく説明する。

【0140】

この画面制御プログラム17におけるレイアウト情報取得機能18は、レイアウト情報のセットを複数回に分けて行うことに対応している。このレイアウト情報取得機能18は、複数に分割されたレイアウト情報19を入力する。

【0141】

図22は、本実施の形態に係る画面制御プログラム17がレイアウト情報を読み込んだ場合の処理手順を例示する図である。

【0142】

処理2201～2205は、先の図3における処理301～305と同様である。これにより制御機能4に対してレイアウト情報が設定され、画面にはフィールドが初期状態で表示される。

【0143】

次に、分割された別のレイアウト情報を指定してレイアウト情報の追加の指示がアプリケーション11よりレイアウト情報取得機能18に対して行われる（206）。

【0144】

レイアウト情報取得機能18は、処理2202と同様の手順でフィールドの属性情報を展開し（2207）、制御機能4に対してレイアウト情報の追加を指示する（2208）。

【0145】

制御機能4は、追加されたフィールドのオブジェクトを作成し、属性をセットする（2209）。以下、処理2207～2208が追加されたフィールドの数だけ繰り返される（2210）。以上により必要に応じたレイアウトの初期化が可能になる。

本実施の形態に係る画面制御プログラム17を利用して、上記第1の実施の

形態と同様の作用効果を得ることができる。

【0146】

(第3の実施の形態)

本実施の形態においては、レイアウト情報、動的表示情報がネットワーク上に配置された環境で利用される画面制御プログラムについて説明する。

【0147】

図23は、本実施の形態に係る画面制御プログラムが計算機システム上で実現する機能を示すブロック図であり、図1と同一の部分については同一の符号をしてその説明を省略し、ここでは異なる部分についてのみ詳しく説明する。

【0148】

画面制御プログラム20を備える計算機21は、ネットワーク22を介してWWWサーバ23と接続されている。

【0149】

レイアウト情報取得機能24は、レイアウト情報9をネットワーク22からダウンロードすることに対応している。

【0150】

動的表示情報取得機能25は、動的表示情報10をネットワーク22からダウンロードすることに対応している。

【0151】

WWWサーバ23は、レイアウト情報9、動的表示情報10のダウンロードを可能とするサーバプロセスの具体例である。

【0152】

レイアウト情報9をネットワーク22からダウンロードする場合には、先の図3に示す処理301においてレイアウト情報9をURL (Uniform Resource Locator) で指定する。

【0153】

レイアウト情報取得機能24は、WWWサーバ23にアクセスして指定されたURLへのコネクションを確立し、レイアウト情報9を読み込む。

【0154】

動的表示情報10をネットワーク22からダウンロードする場合もレイアウト情報9の場合と同様に行われ、先の図10に示す処理1001において動的表示情報10をURLで指定する。

【0155】

Java言語ではURLで指定されたファイルを扱うクラスが用意されており、このクラスを用いることで実現可能である。

【0156】

本実施の形態に係る画面制御プログラム20を利用して、上記第1の実施の形態と同様の作用効果を得ることができる。

【0157】

(第4の実施の形態)

本実施の形態においては、各インターフェースにプラグインされるコンポーネントをネットワーク上に分散させた環境で利用される画面制御プログラムについて説明する。

【0158】

図24は、本実施の形態に係る画面制御プログラムが計算機システム上で実現する機能を示すブロック図であり、図1と同一の部分については同一の符号をしてその説明を省略し、ここでは異なる部分についてのみ詳しく説明する。

【0159】

画面制御プログラム26は、図示しない計算機上で動作する。画面制御プログラム26における検索処理インターフェース27は、検索処理タブ28と接続されている。

【0160】

この検索処理タブ28は、検索処理インターフェース27にプラグインされる検索処理用のコンポーネントであり、ネットワーク29を介して検索処理リモートオブジェクト30に検索処理を依頼し、検索処理リモートオブジェクト30が行った検索処理結果を受け取り、検索処理インターフェース27に返却する。

【0161】

同様に、入力チェック処理インターフェース31は、入力チェック処理タブ3

2と接続されている。

【0162】

この入力チェック処理スタブ32は、入力チェック処理インターフェース31にプラグインされる入力チェック処理用のコンポーネントであり、ネットワーク29を介して入力チェック処理リモートオブジェクト33に入力チェック処理を依頼し、入力チェック処理リモートオブジェクト33が行った入力チェック処理結果を受け取り、入力チェック処理インターフェース31に返却する。

【0163】

本実施の形態に係る画面制御プログラム26を利用して、上記第1の実施の形態と同様の作用効果を得ることができる。

【0164】

(第5の実施の形態)

本実施の形態においては、画面表示トランザクション情報がサーバに保持され、クライアントの状態をサーバから制御する画面制御プログラムについて説明する。

【0165】

図25は、本実施の形態に係る画面制御プログラムが計算機システム上で実現する機能を示すブロック図であり、図1と同一の部分については同一の符号を付してその説明を省略し、ここでは異なる部分についてのみ詳しく説明する。

【0166】

画面制御プログラム34は、図示しないクライアント上で動作する。画面制御プログラム34における画面表示トランザクション機能35は、画面表示トランザクション情報送受信部36に接続されている。

【0167】

画面表示トランザクション情報送受信部36は、画面表示トランザクション機能35から画面表示トランザクション情報（画面の状態情報）を取得し、ネットワーク37を介して画面表示トランザクション管理サーバ38に送信する。また、画面表示トランザクション管理サーバ38から受信した画面表示トランザクション情報を画面表示トランザクション機能35にセットする。

【0168】

画面表示トランザクション管理サーバ38は、画面表示トランザクション送受信部36から受信した画面表示トランザクション情報をクライアント毎に管理し、必要に応じて取出し画面表示トランザクション情報送受信部36に送信する。

【0169】

本実施の形態に係る画面制御プログラム34を利用して、上記第1の実施の形態と同様の作用効果を得ることができる。

【0170】

(第6の実施の形態)

本実施の形態においては、業務処理をプラグインする画面制御プログラムについて説明する。

【0171】

図26は、本実施の形態に係る画面制御プログラムが計算機上で実現する機能を示すブロック図であり、図1と同一の部分については同一の符号を付してその説明を省略し、ここでは異なる部分についてのみ詳しく説明する。

【0172】

画面制御プログラム39の制御機能40は、業務処理インターフェース41と接続されている。

【0173】

業務処理用のコンポーネント42は、他の画面表示に関わるコンポーネントと同様の手法により業務処理インターフェース41にプラグインされる。

【0174】

アプリケーション11は、画面表示に関する処理のみではなく、業務処理もこの画面制御プログラム39を利用して実行可能である。

【0175】

ユーザは、この業務処理用のコンポーネント42の利用を望む場合には、他の画面表示に関わるコンポーネントと同様に、コンポーネント42を指定し、必要な属性を指定するのみでよい。

【0176】

これにより、画面プログラムの開発のみではなく、アプリケーションの開発も効率化できる。

【0177】

なお、上記各実施の形態で説明した発明は、それぞれを自由に組み合わせることができる。また、上記各実施の形態で説明した各機能、各要素は、同様の作用・機能を実現可能であれば配置を変更させてもよく、また各機能、各要素を自由に組み合わせててもよい。

【0178】

上記各実施の形態で説明した画面制御プログラムの各機能、各要素は、コンピュータに実行させることのできるプログラムであり、例えば磁気ディスク（フロッピーディスク、ハードディスク等）、光ディスク（CD-ROM、DVD等）、半導体メモリなどの記録媒体に書き込んで適用したり、通信媒体により伝送して、計算機、計算機システムに適用可能である。

【0179】

上記各実施の形態で説明した画面制御プログラムの各機能を実現するコンピュータは、記録媒体に記録されたプログラムを読み込み、プログラムによって動作が制御されることにより、上述した処理を実行する。

【0180】

【発明の効果】

以上詳記したように本発明においては、動的に変化する表示内容を得るために予め作成されている動的表示処理を指定し、この動的表示処理への属性を指定するのみで、動的な表示内容を取得し画面に表示させることができる。

【0181】

したがって、画面プログラムの開発や変更においてプログラミングする必要がなく、コンパイルする必要もない。ゆえに、ユーザは画面プログラムを効率的かつ容易に開発・変更でき、ユーザの労力が軽減される。

【0182】

また、本発明においては、正常な入力を保持しておき、障害発生後にはこの保持しておいた内容に基づいて、入力内容が再現される。

【0183】

したがって、ユーザの操作が効率化され、容易に画面プログラムの耐障害性を向上させることができる。

【図面の簡単な説明】

【図1】

本発明の第1の実施の形態に係る画面制御プログラムが計算機上で実現する機能を示すブロック図。

【図2】

レイアウト情報の具体例を示す図。

【図3】

同実施の形態に係る画面制御プログラムがレイアウト情報を読み込んだ場合の処理手順を例示する図。

【図4】

同実施の形態において <field> タグの数だけ繰り返される各フィールドの作成処理を例示するフローチャート。

【図5】

レイアウト情報に基づく制御機能の処理によって構成される画面を例示する図

【図6】

同実施の形態において画面制御プログラムがレイアウト情報を読み込んだ場合の処理手順の変形例を示す図。

【図7】

レイアウト情報において定義された入力チェック属性に従って画面制御プログラムが実行する入力チェック処理の処理手順を例示する図。

【図8】

入力チェックの結果がエラーの場合の画面を例示する図。

【図9】

動的表示情報の具体例を示す図。

【図10】

同実施の形態に係る画面制御プログラムが動的表示情報を読み込んだ場合の処理手順を例示する図。

【図11】

動的表示情報に基づく制御機能の処理によって構成される画面を例示する図。

【図12】

同実施の形態において画面制御プログラムが動的表示情報を読み込んだ場合の処理手順の変形例を示す図。

【図13】

同実施の形態においてフィールドにセットされた表示属性が検索だった場合に画面制御プログラムが実行する処理の手順を例示する図。

【図14】

同実施の形態においてフィールドにセットされた表示属性が検索だった場合の画面制御プログラムの処理を例示するフローチャート。

【図15】

同実施の形態において検索処理用のコンポーネントによって動的な表示内容が取得され、検索された一覧がユーザの操作に基づいて表示された画面を例示する図。

【図16】

同実施の形態においてフィールドにセットされた表示属性がポップアップだった場合に画面制御プログラムが実行する処理の手順を例示する図。

【図17】

同実施の形態において、ユーザがフィールドに対して操作を行った結果、起動されるポップアップ処理の手順を例示する図。

【図18】

ポップアップウィンドウが表示されている画面を示す図。

【図19】

ポップアップウィンドウで選択されたデータがフィールドにセットされた画面を示す図。

【図20】

本実施の形態において画面表示トランザクション機能を利用して画面表示を復帰させる処理の手順を例示する図。

【図21】

本発明の第2の実施の形態に係る画面制御プログラムが計算機上で実現する機能を示すブロック図。

【図22】

同実施の形態に係る画面制御プログラムがレイアウト情報を読み込んだ場合の処理手順を例示する図。

【図23】

本発明の第3の実施の形態に係る画面制御プログラムが計算機上で実現する機能を示すブロック図。

【図24】

本発明の第4の実施の形態に係る画面制御プログラムが計算機システム上で実現する機能を示すブロック図。

【図25】

本発明の第5の実施の形態に係る画面制御プログラムが計算機システム上で実現する機能を示すブロック図。

【図26】

本発明の第6の実施の形態に係る画面制御プログラムが計算機上で実現する機能を示すブロック図。

【符号の説明】

1、17、20、26、34、39…画面制御プログラム

2、18、24…レイアウト情報取得機能

3、25…動的表示情報取得機能

3 a…取得機能

3 b…変換機能

3 c…提供機能

4、40…制御機能

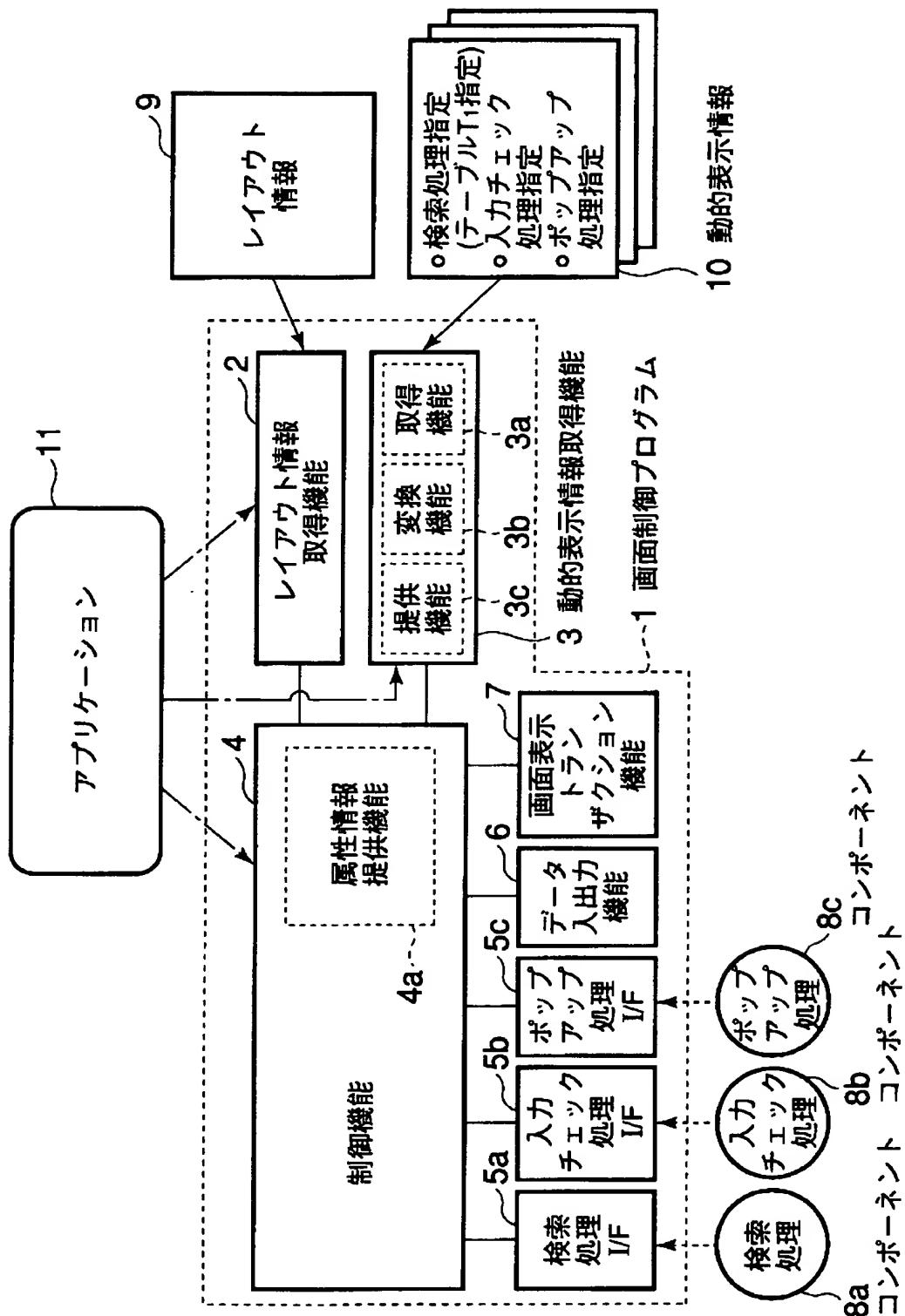
4 a…属性情報提供機能

5 a、27…検索処理インターフェース
5 b、31…入力チェック処理インターフェース
5 c…ポップアップ処理インターフェース
6…データ入出力機能
7、35…画面表示トランザクション機能
8 a～8 c…コンポーネント
9、19…レイアウト情報
10…動的表示情報
11…アプリケーション
21…計算機
22、29、37…ネットワーク
23…WWWサーバ
28…検索処理スタブ
30…検索処理リモートオブジェクト
32…入力チェック処理スタブ
33…入力チェック処理リモートオブジェクト
36…画面表示トランザクション情報送受信部
38…画面表示トランザクション管理サーバ
41…業務処理インターフェース
42…業務処理

【書類名】

四面

【図1】



【図2】

9 レイアウト情報

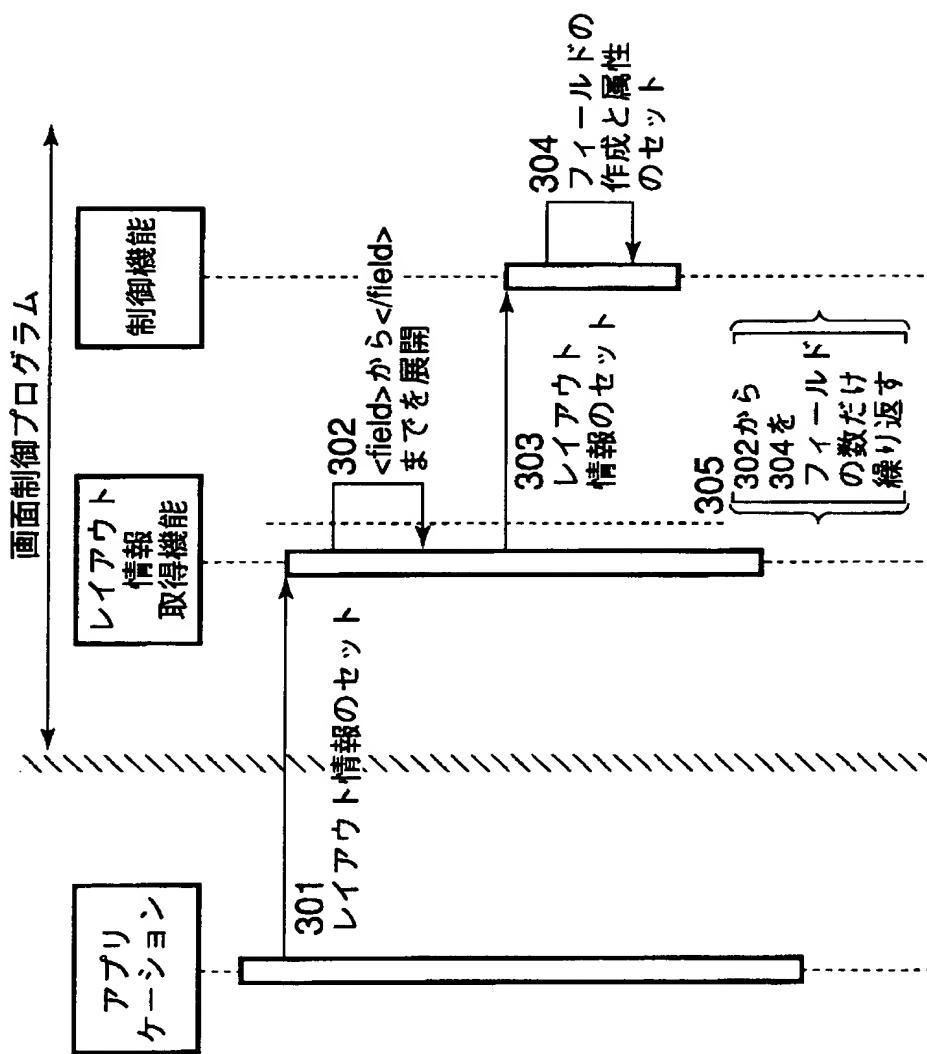
```

1 <layout>
2   <field id="order"> ----- フィールドのID
3     <x>5</x>
4     <y>5</y>
5     <width>300</width>
6     <height>20</height>
7     <label>注文番号</label>
8     <buttonlabel>参照</buttonlabel>
9     <fieldstringnumber>8</fieldstringnumber>
10    <labelstringnumber>8</labelstringnumber>
11    <buttonstringnumber>4</buttonstringnumber>
12    <initialstate>UNEDITABLE</initialstate>
13    <validation>
14      <trigger>VK_ENTER</trigger>
15      <component>order.NumericCheckComponent</component>
16      <behaviour>
17        <type>CHANGE_BACKGROUND_COLOR</type>
18        <param name="background">#FF0000</param>
19      </behaviour>
20      <validation>
21    </field>
22    <field id="area">
23      <x>5</x>
24      <y>25</y>
25      <width>300</width>
26      <height>20</height>
27      <label>地域</label>
28      <buttonlabel>参照</buttonlabel>
29      <fieldstringnumber>10</fieldstringnumber>
30      <labelstringnumber>8</labelstringnumber>
31      <buttonstringnumber>4</buttonstringnumber>
32      <initialstate>UNEDITABLE</initialstate>
33    </field>
34  ..
35 </layout>

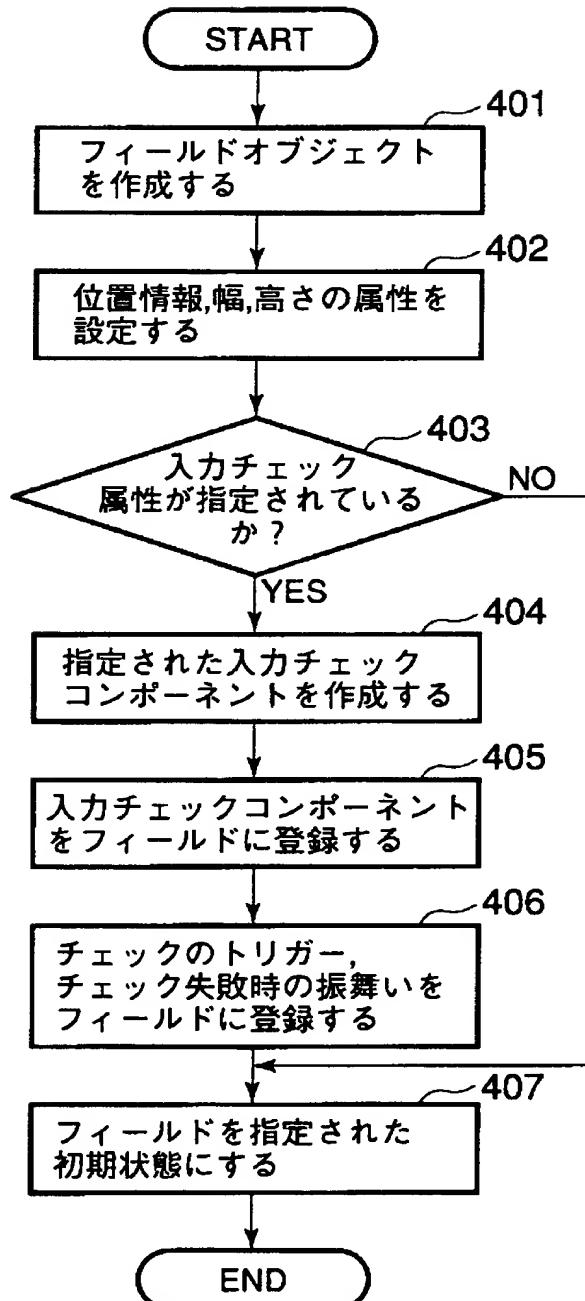
```

{ フィールド1つに
関する定義 }

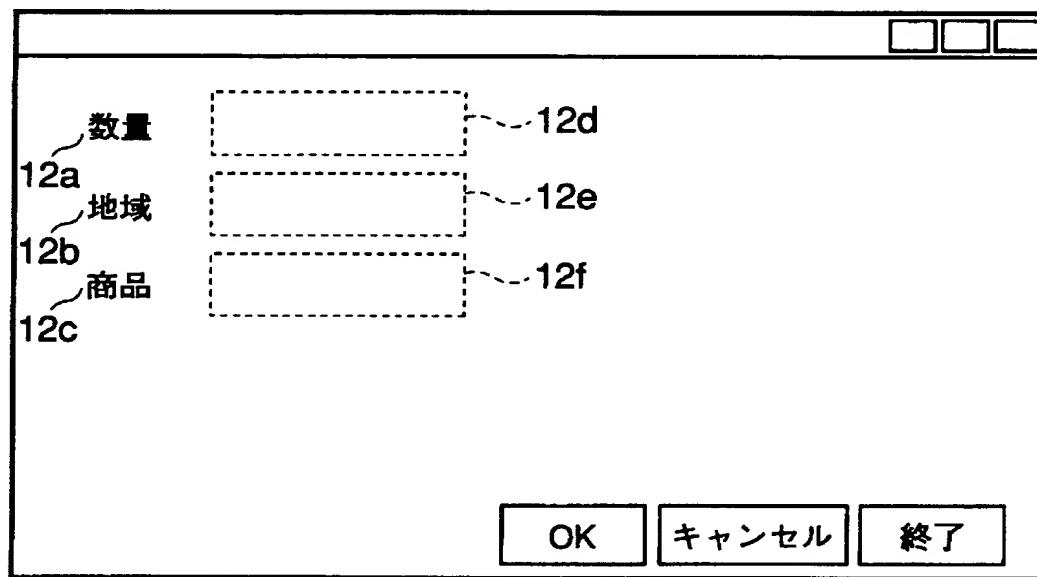
【図3】



【図4】

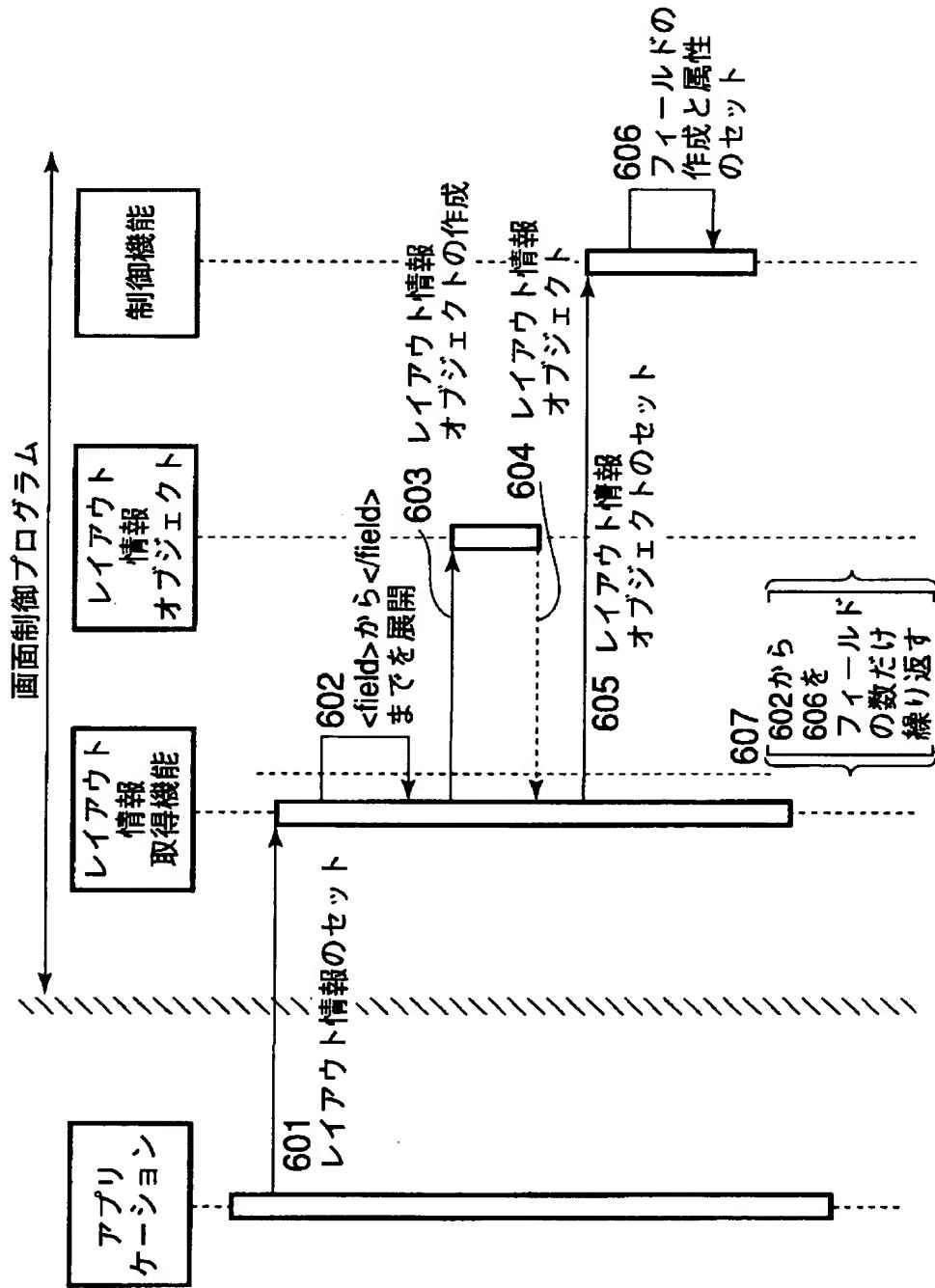


【図5】

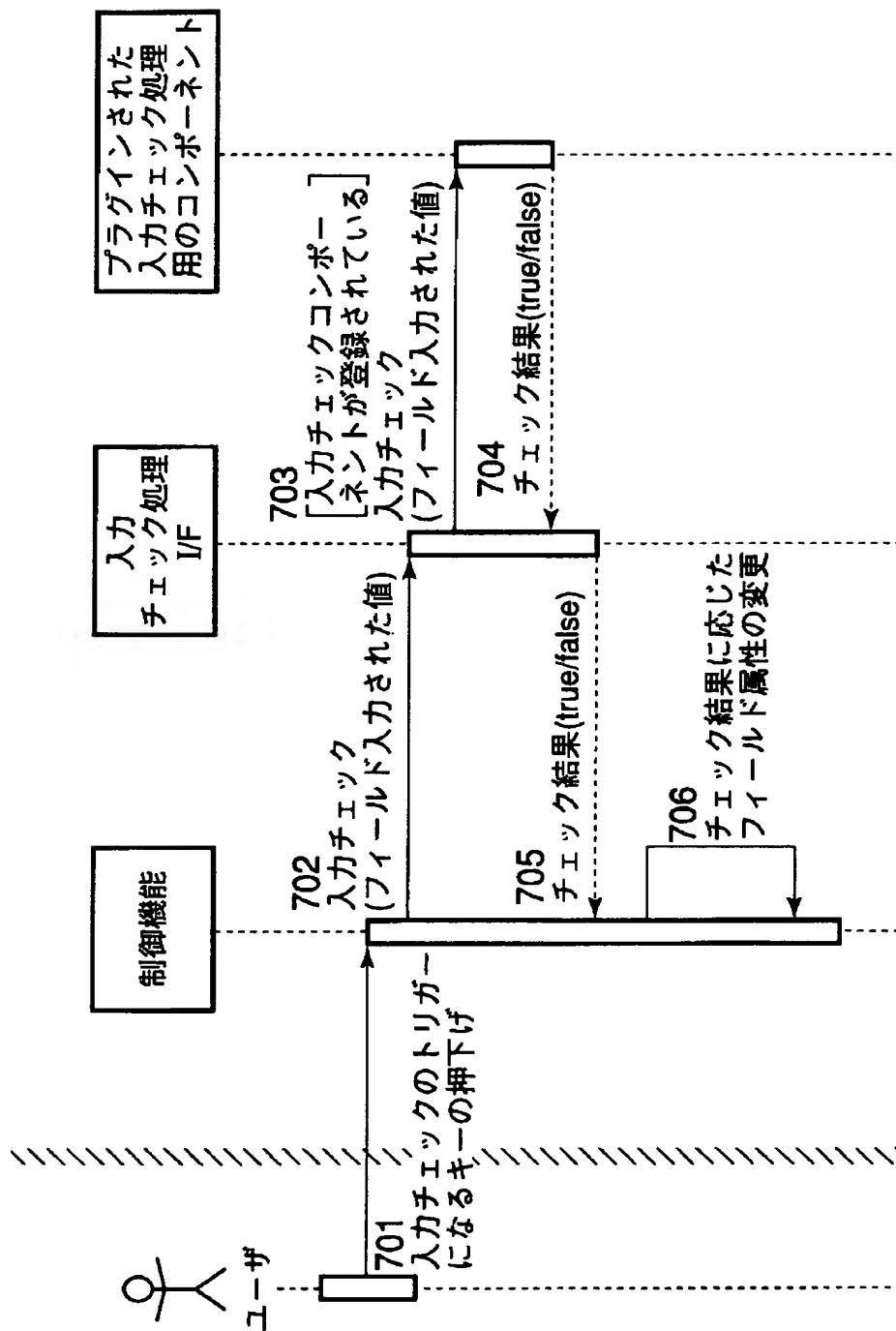


12 画面

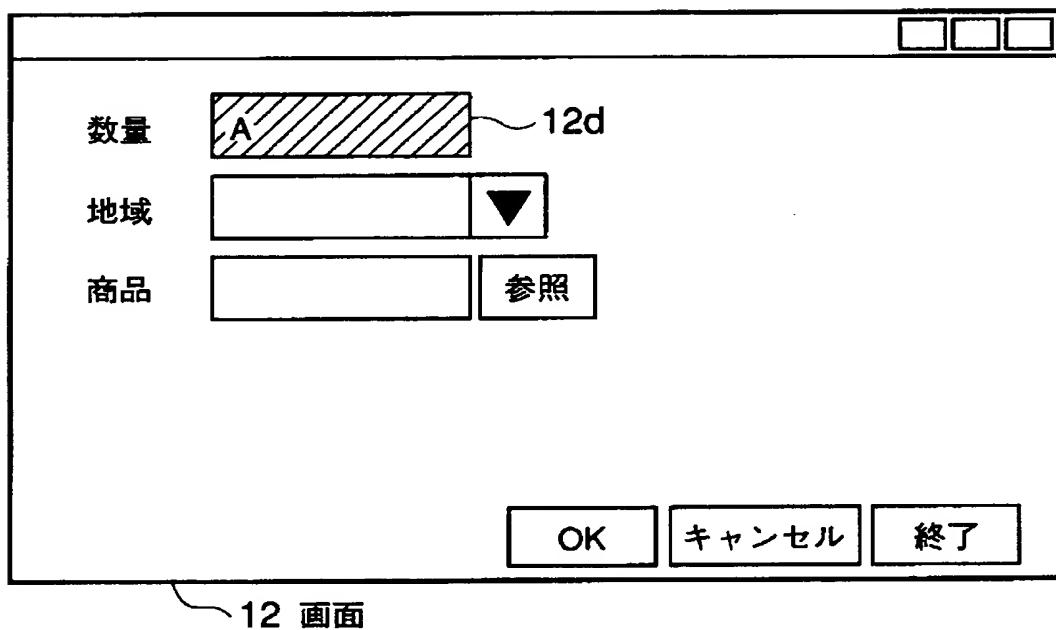
【図6】



【図7】



【図8】



【図9】

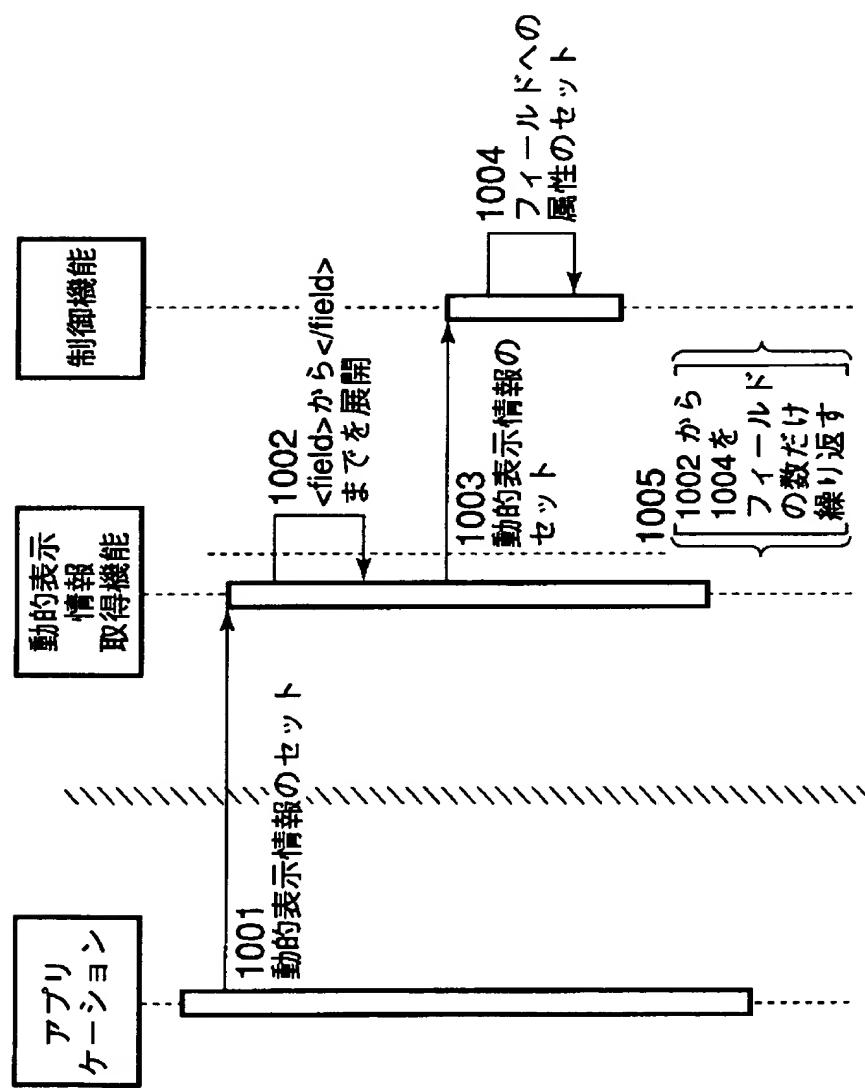
10
動的表示情報

```

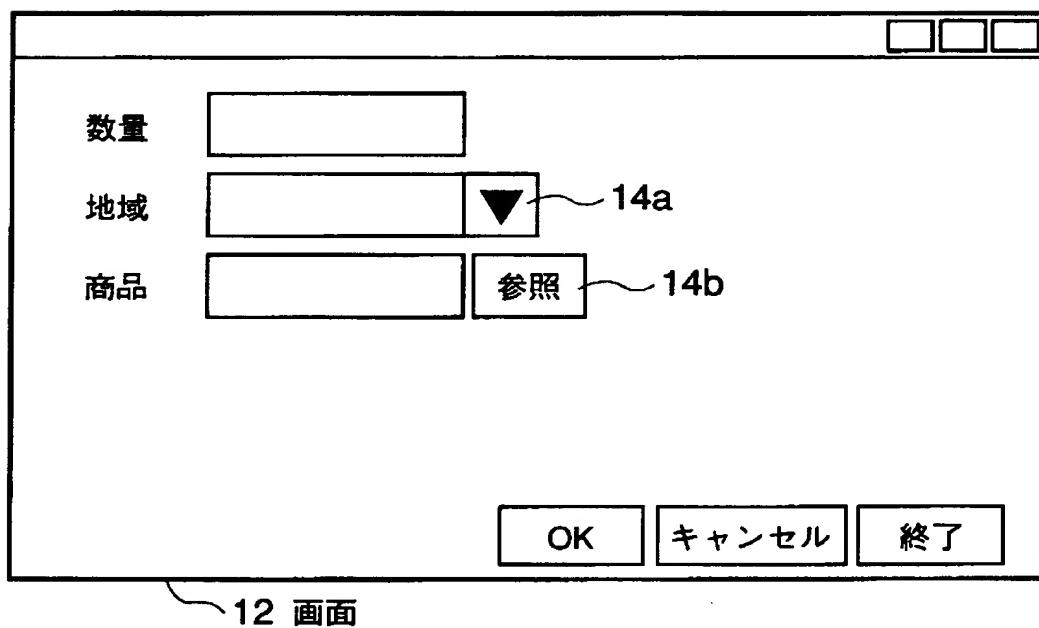
1 <pattern>
2   <field id="area">
3     <state>COMBO</state>———— フィールドの表示状態
4   <search>
5     <table><area_master></table>
6       <column>code</column>
7       <column>name</column>
8       <condition>
9         <column>charge</column>
10        <value>1000</value>
11      </condition>
12    </search>
13  </field>
14  <field id="product">
15    <state>POPUP</state>
16    <popup>
17      <panel>order.ProductPanel</panel>
18        <title>商品検索</title>
19        <width>250</width>
20        <height>300</height>
21        <okbuttonlabel>OK</okbuttonlabel>
22        <cancelbuttonlabel>キャンセル</cancelbuttonlabel>
23    </popup>
24  </field>
25 ...
26 </pattern>

```

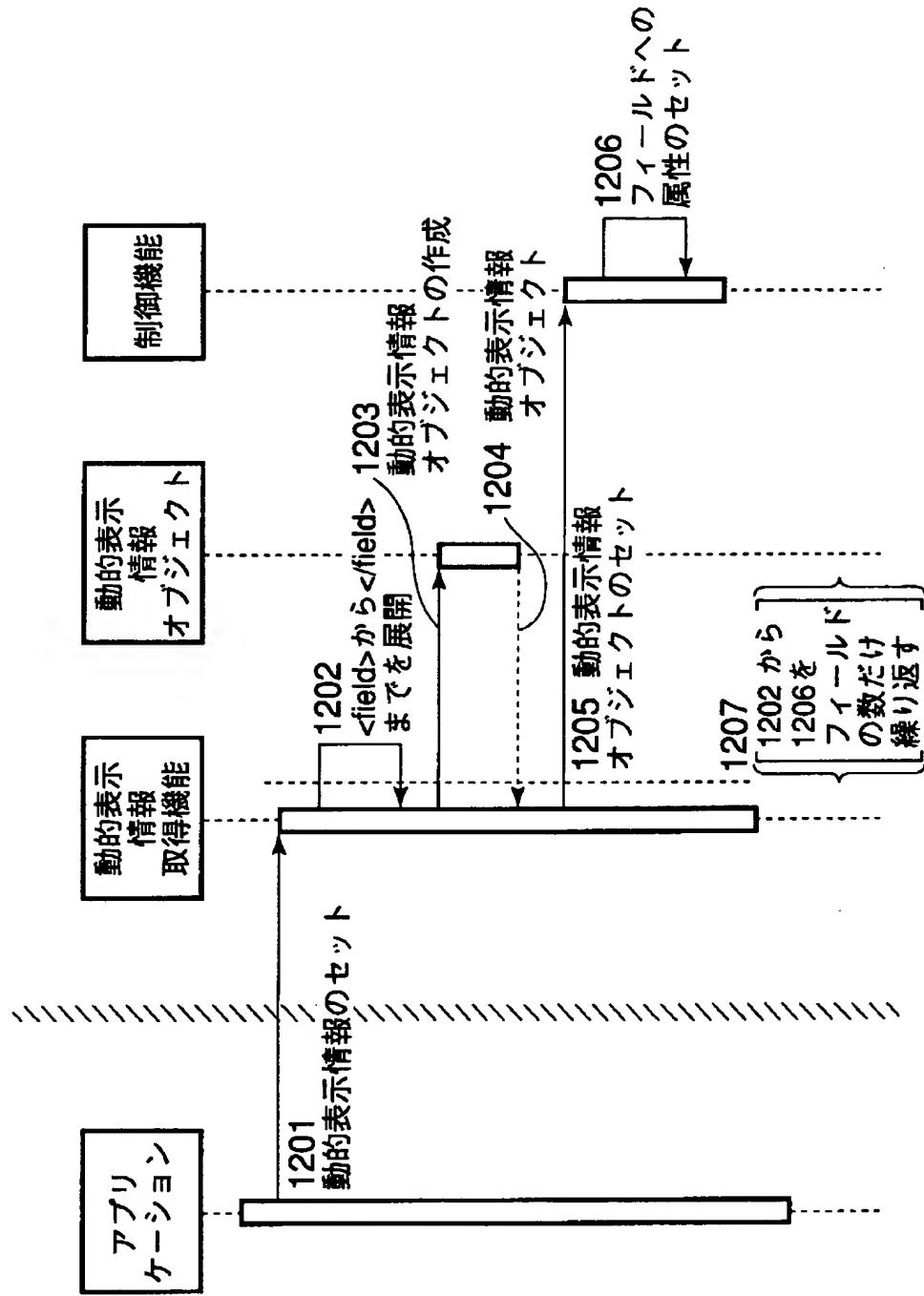
【図10】



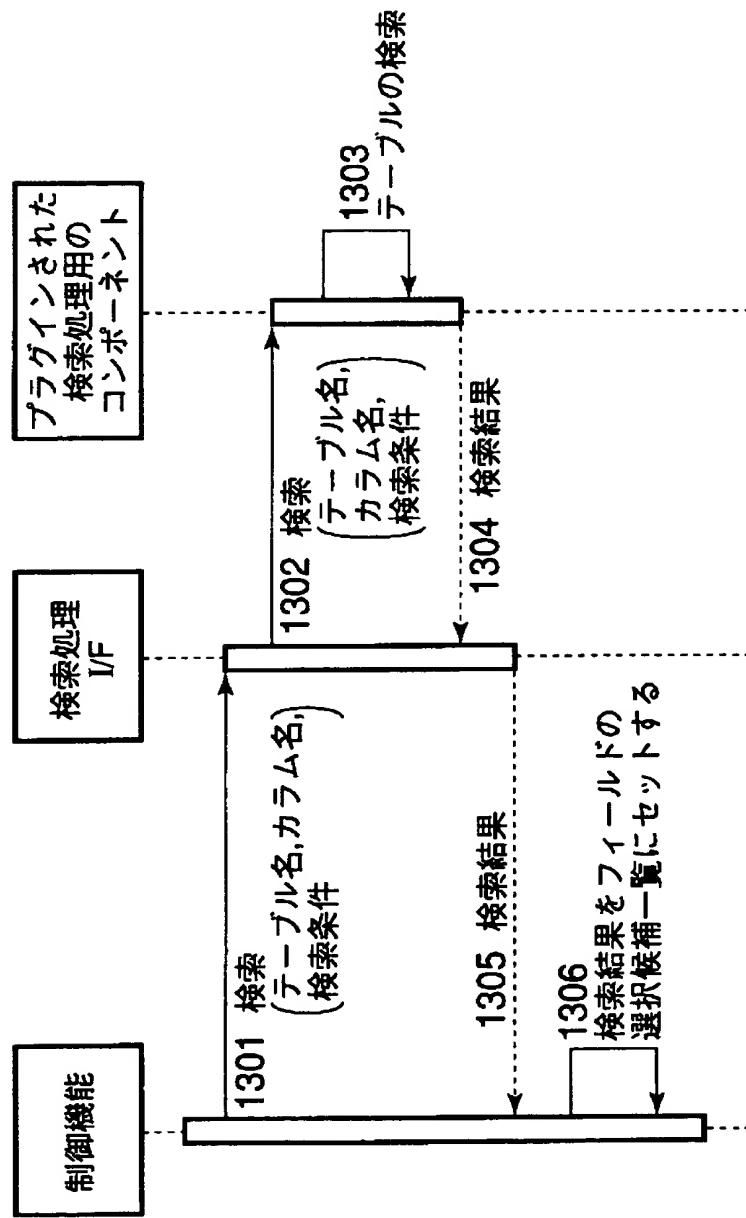
【図11】



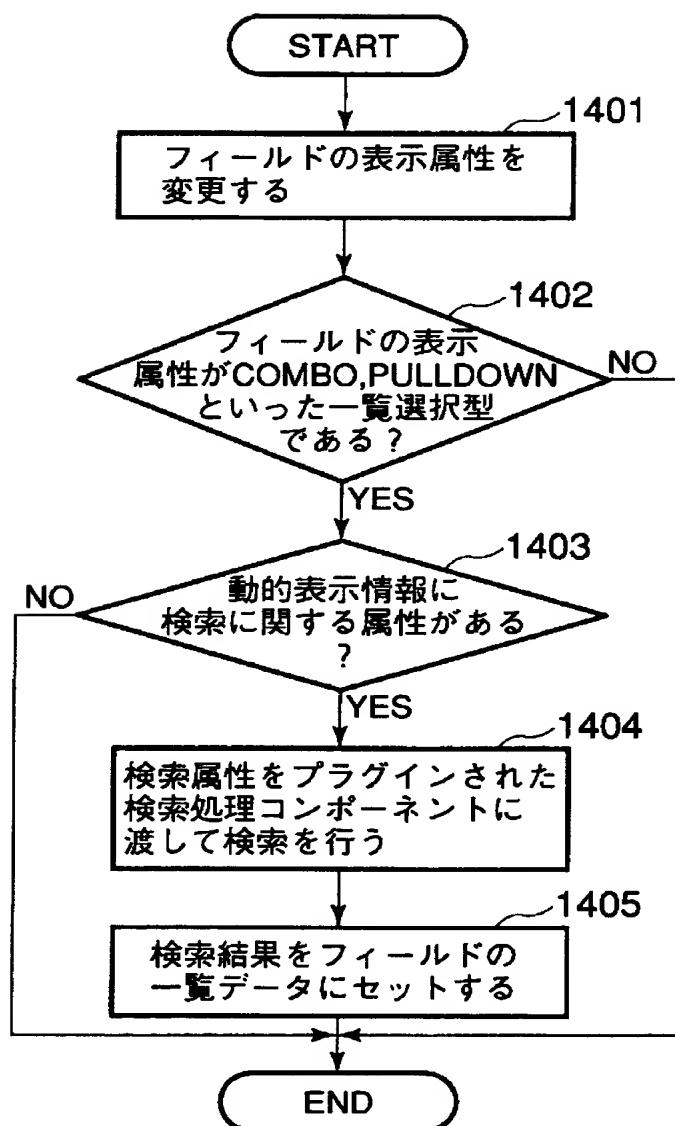
【図12】



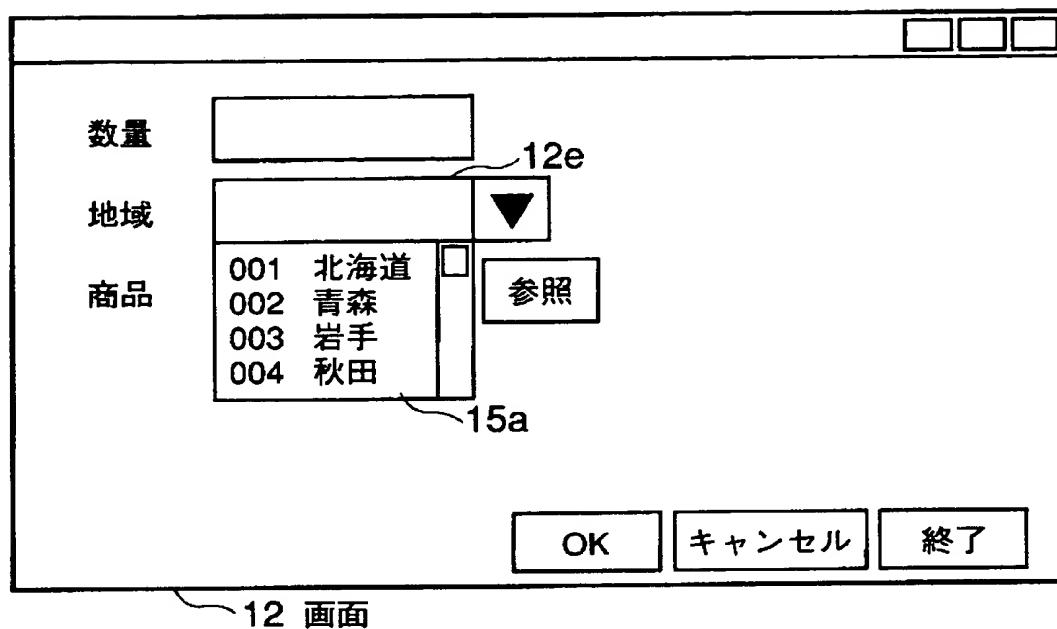
【図13】



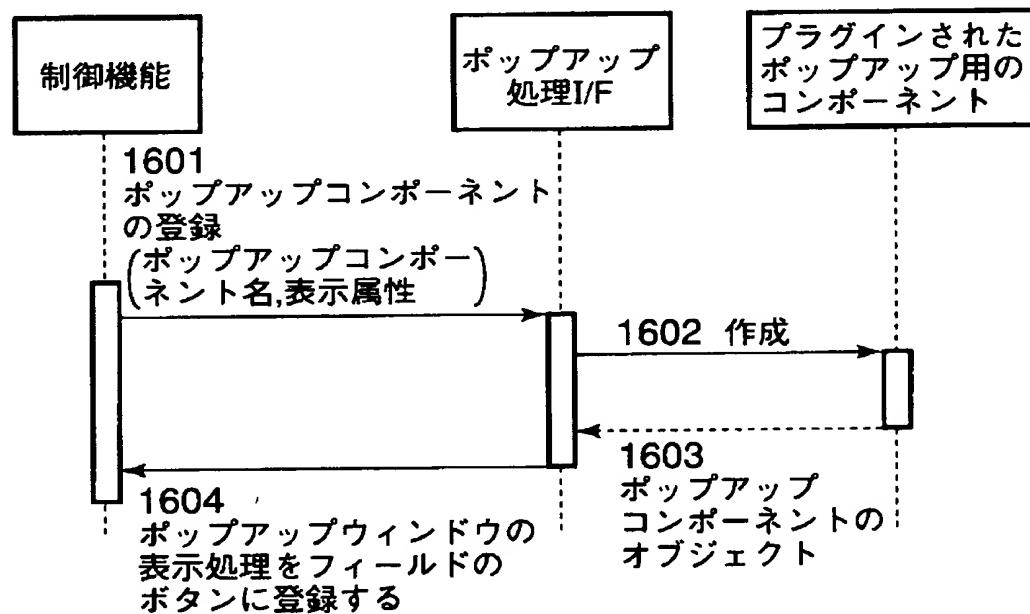
【図14】



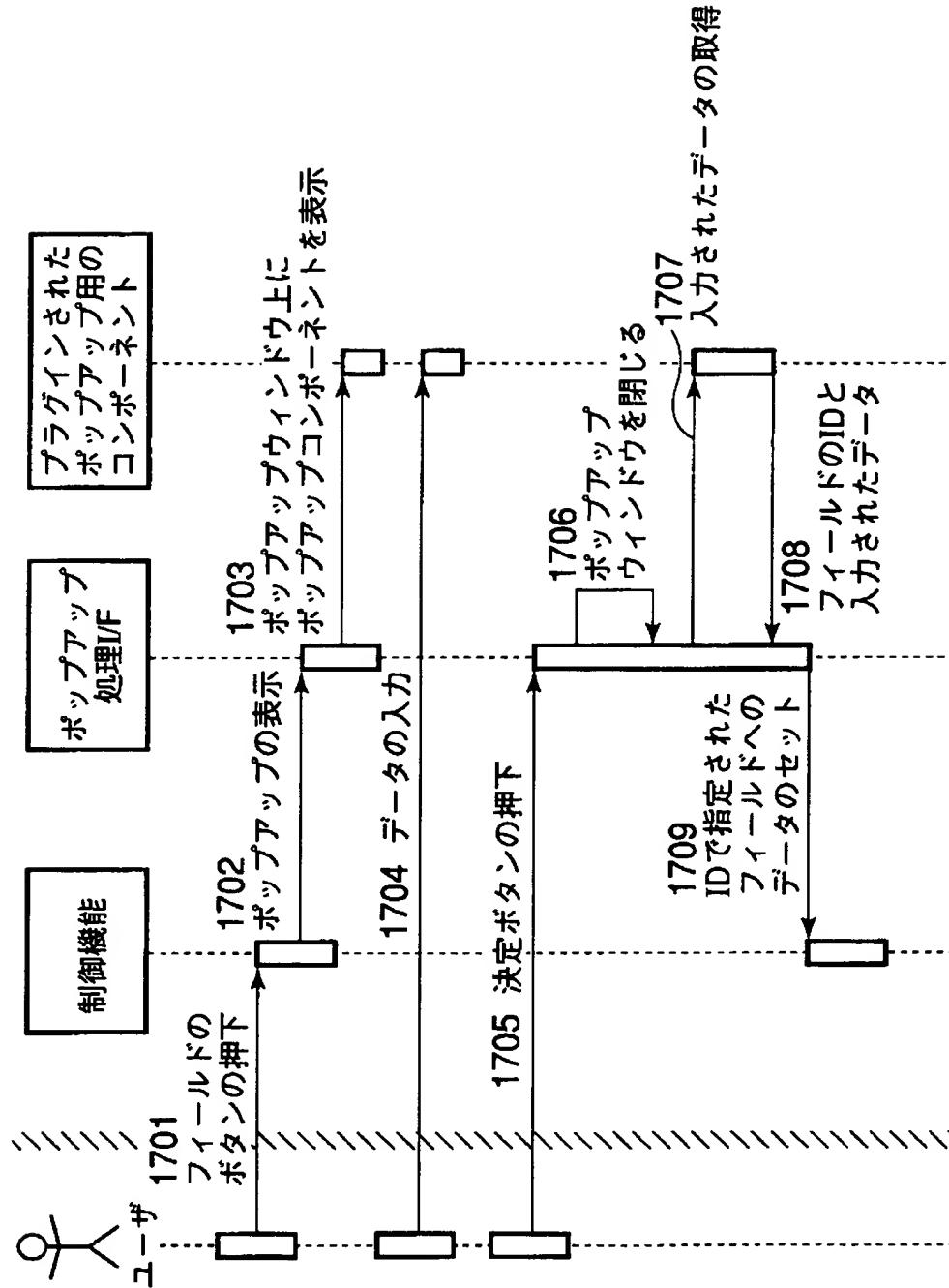
【図15】



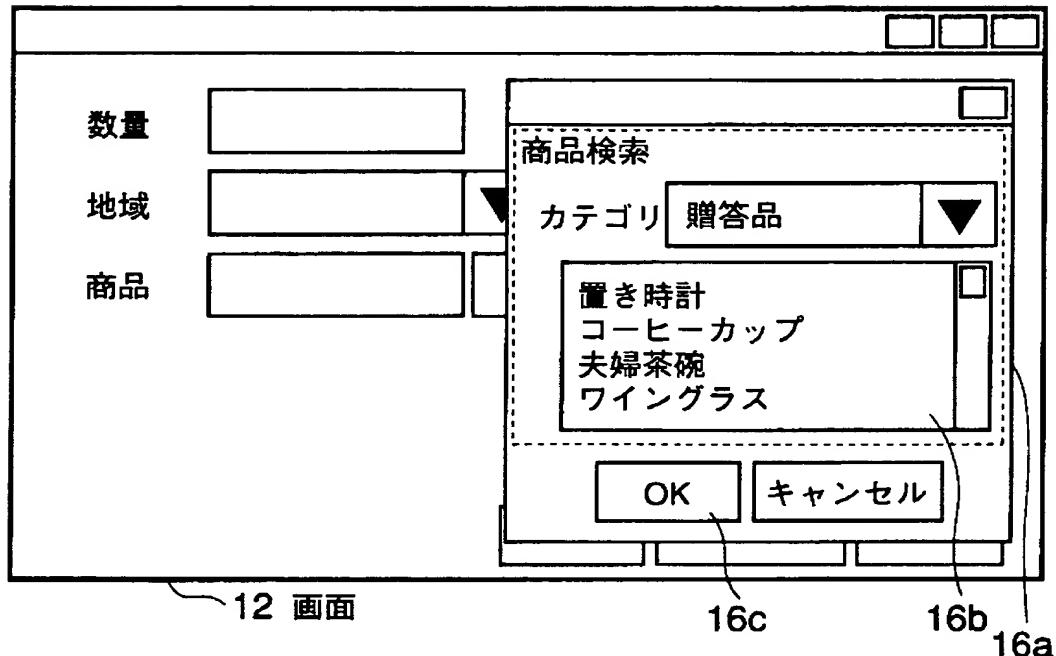
【図16】



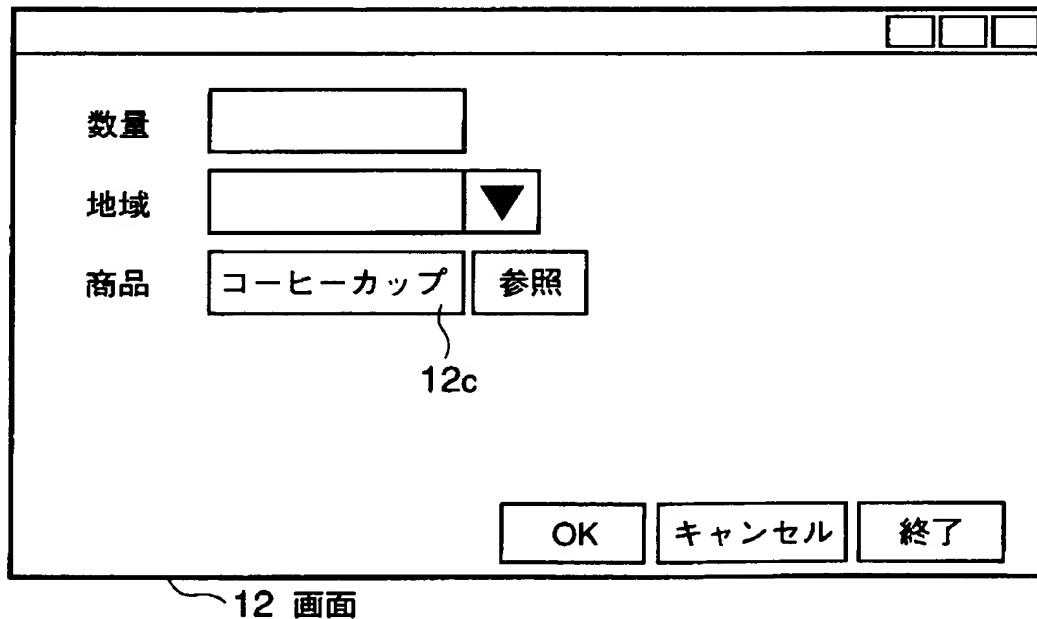
【図17】



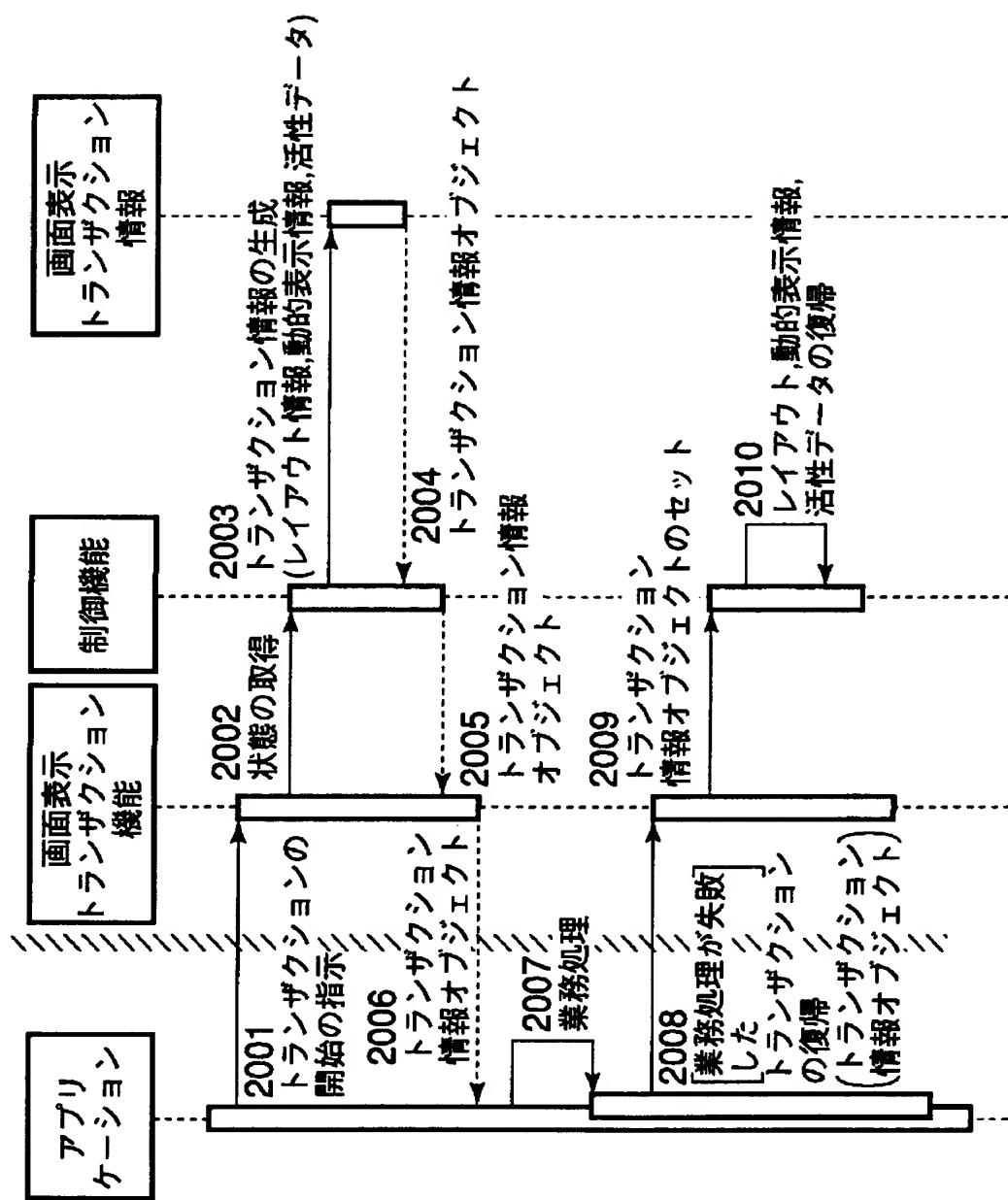
【図18】



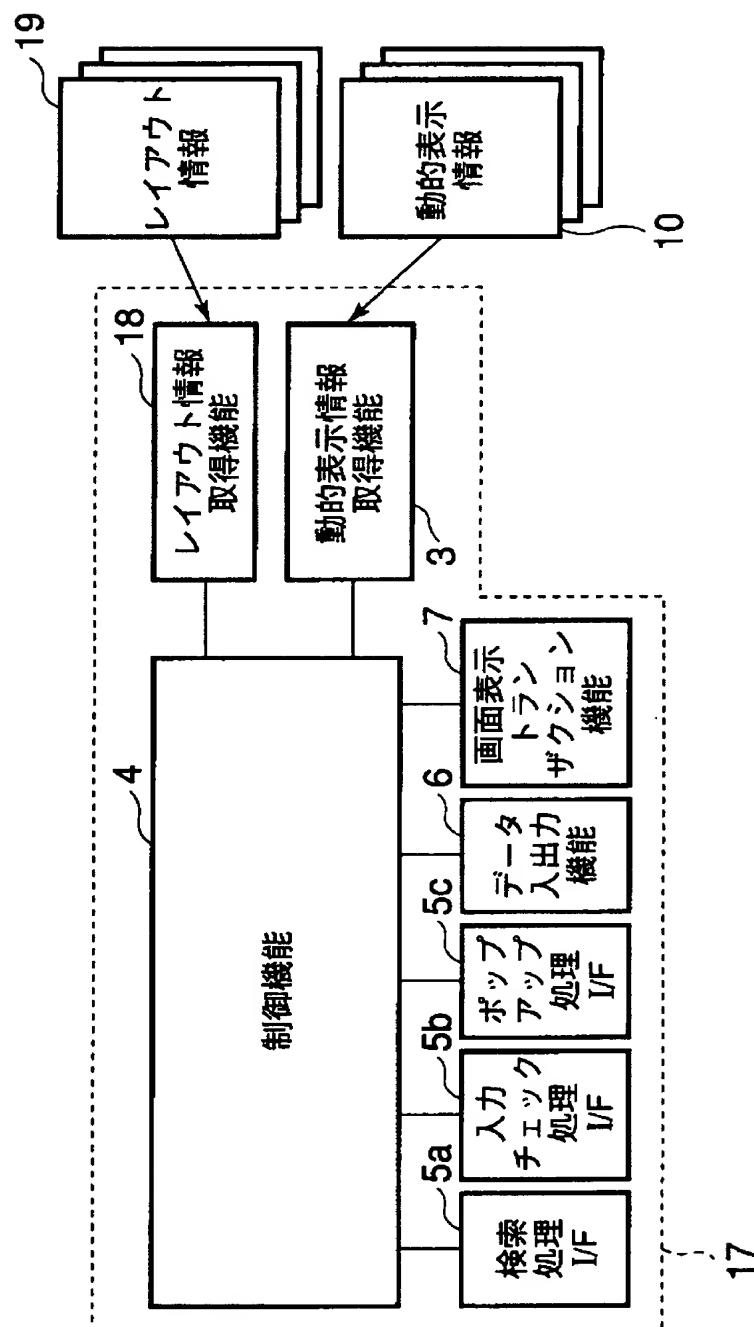
【図19】



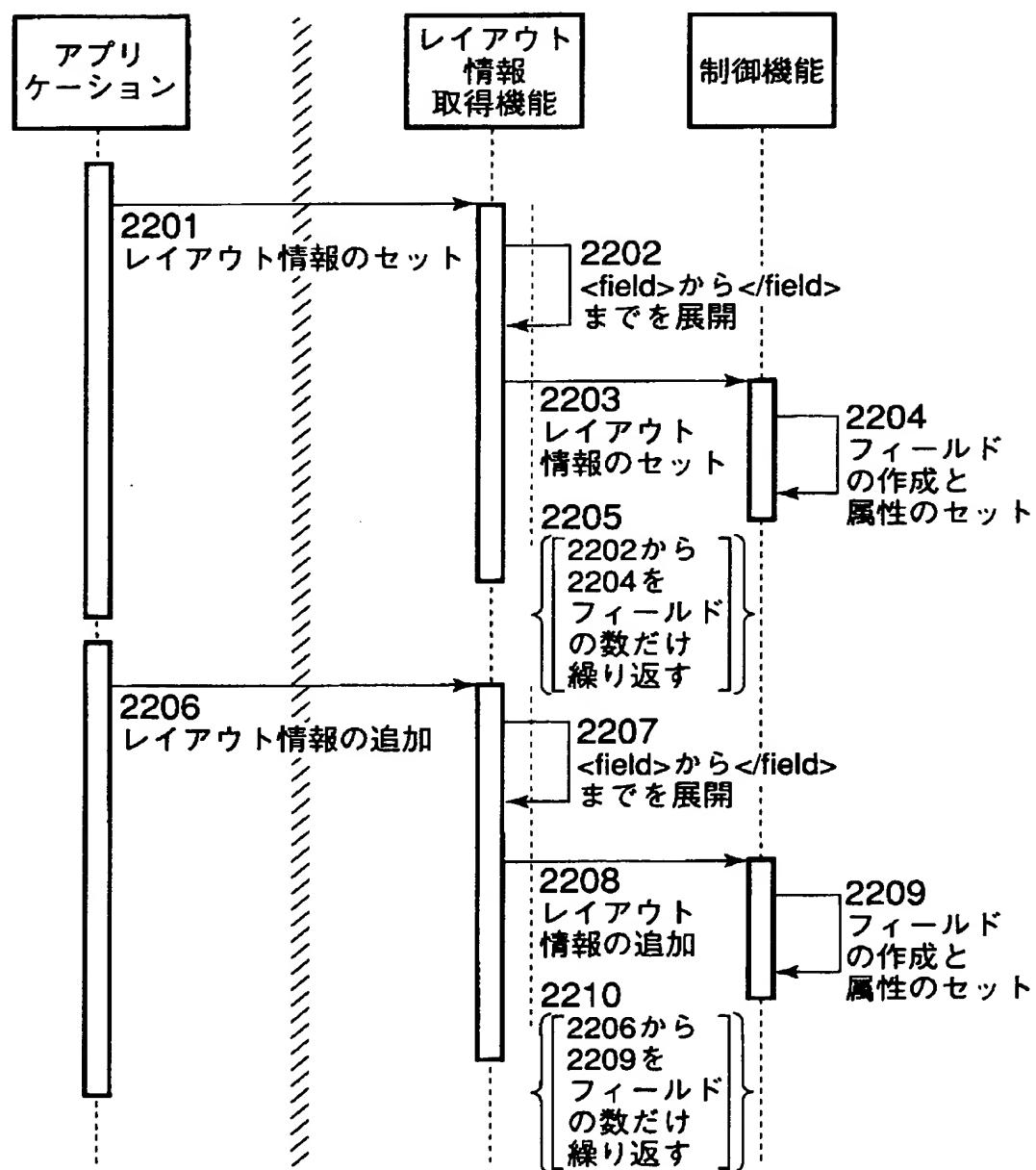
【図20】



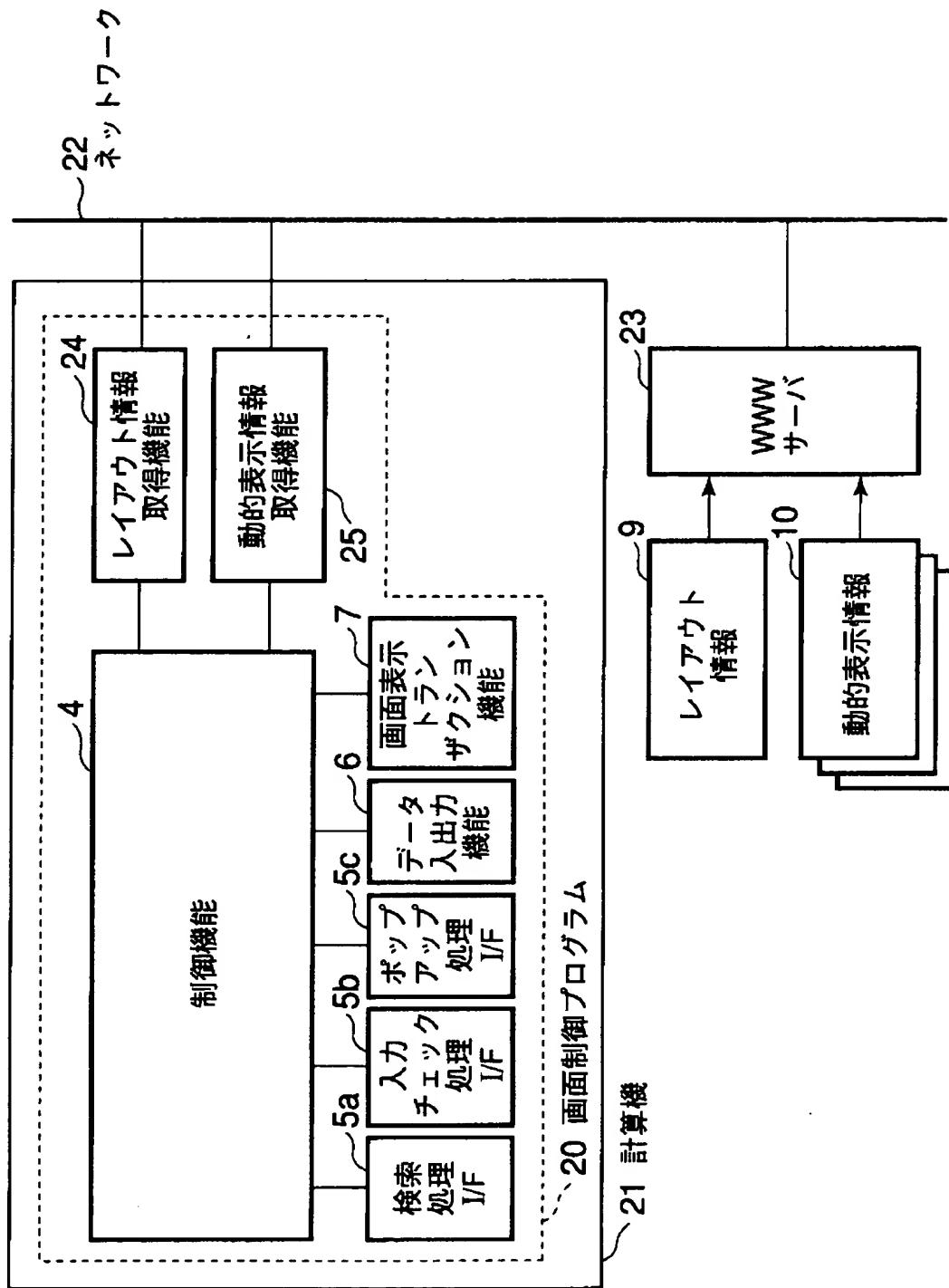
【図21】



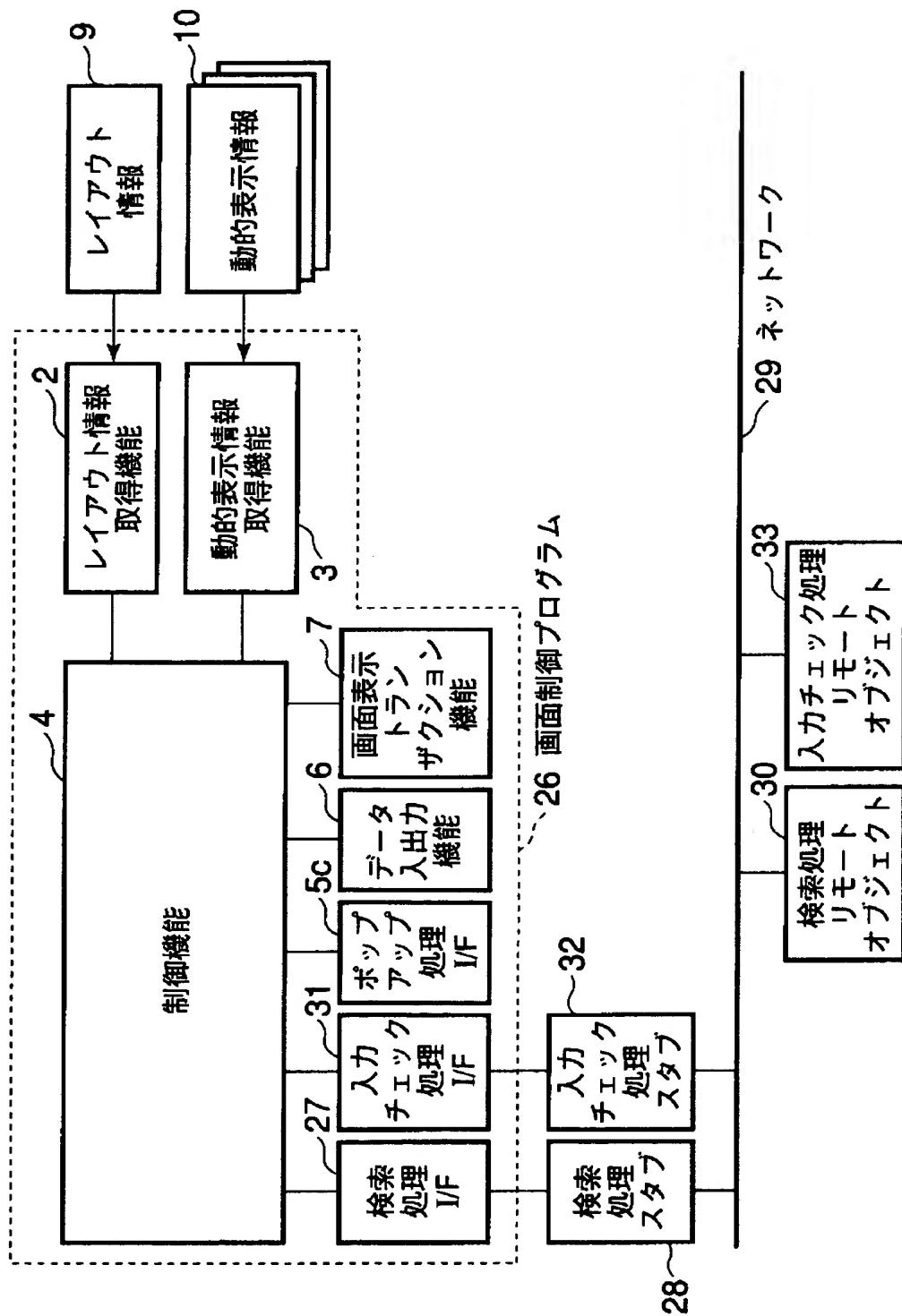
【図22】



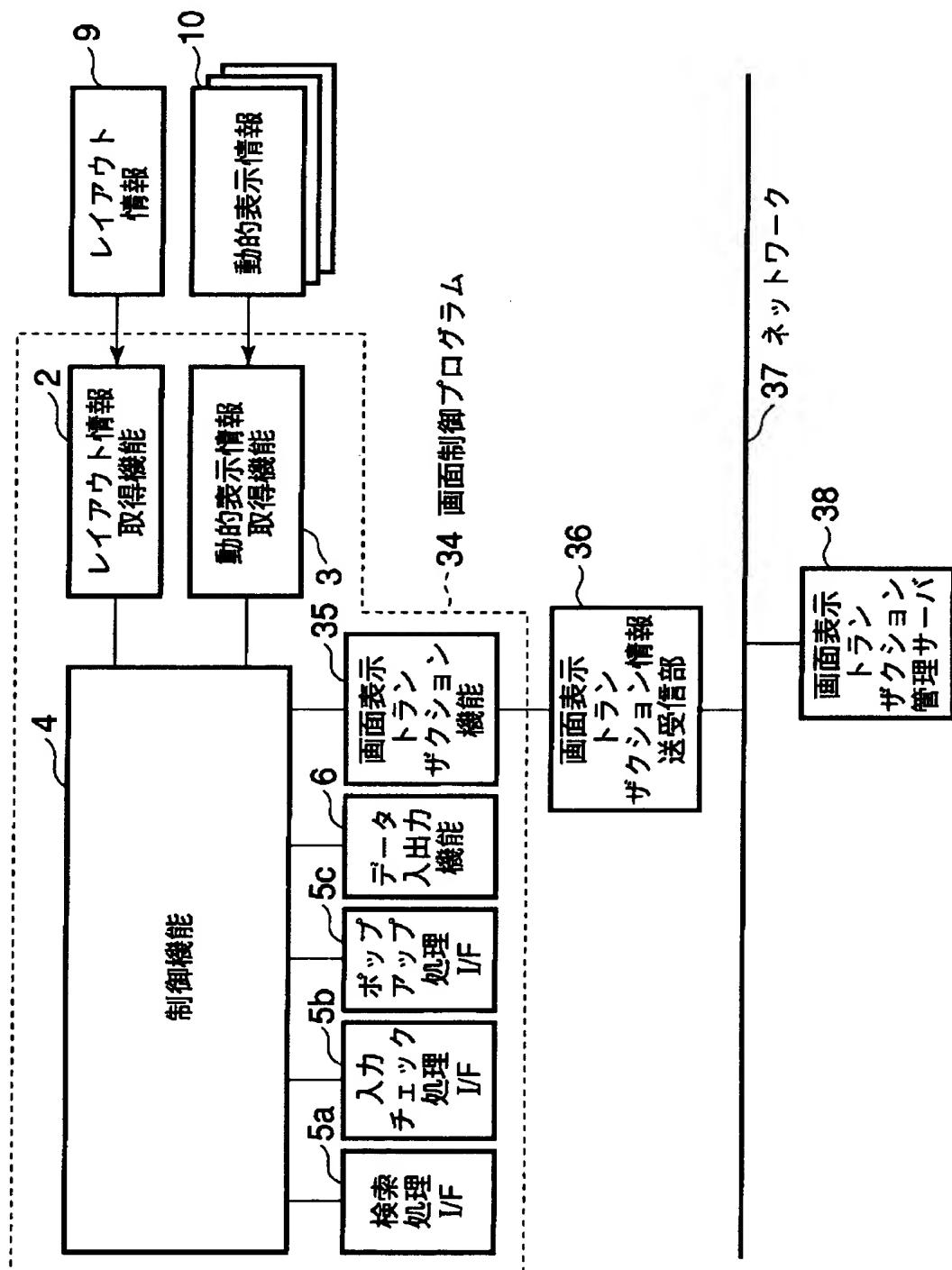
【図23】



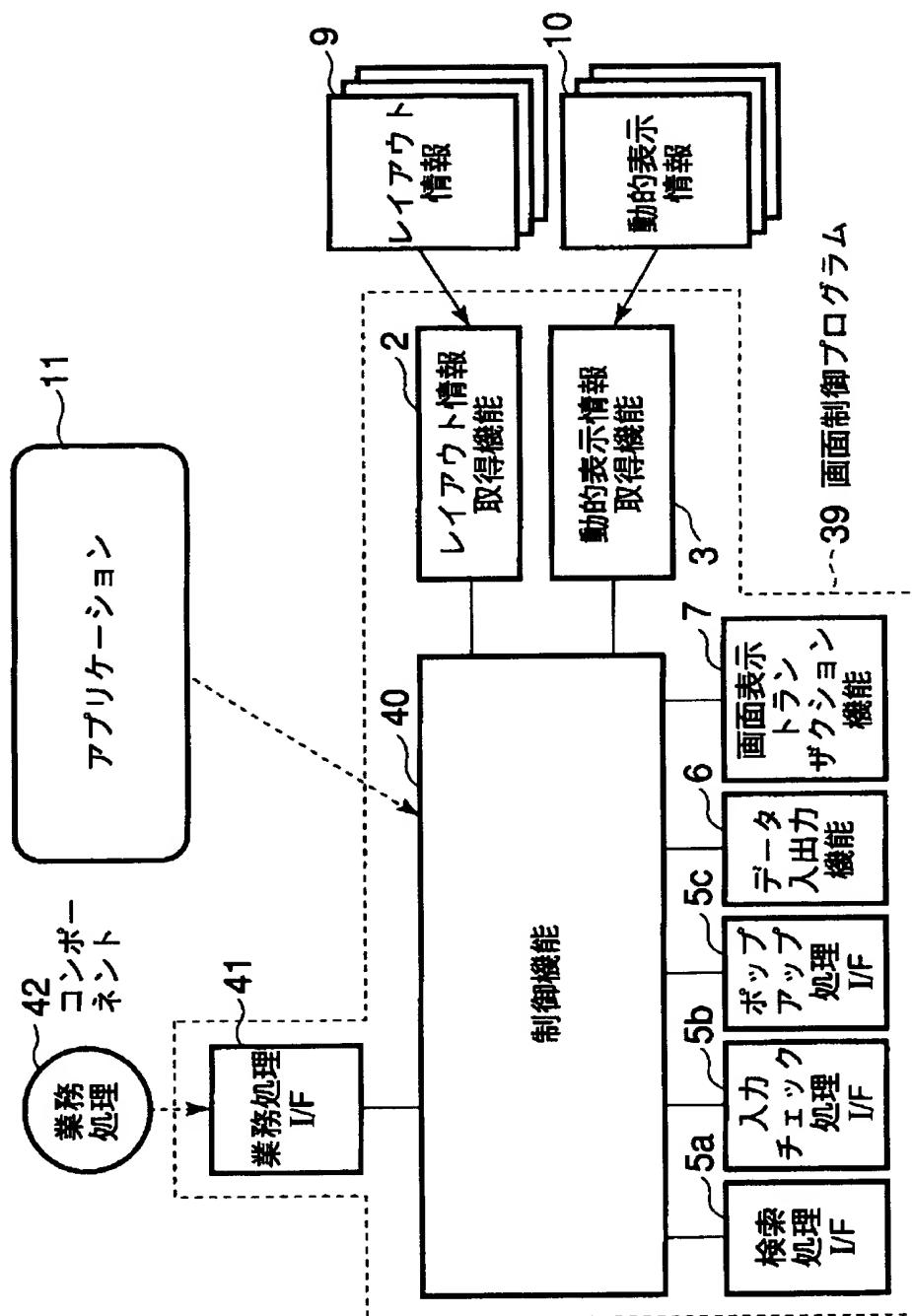
【図24】



【図25】



【図26】



【書類名】 要約書

【要約】

【課題】 画面プログラムの開発・変更を効率化する。

【解決手段】 コンピュータに、画面に表示される場合に動的に変化する表示内容を得るために予め作成されている少なくとも一つの動的表示処理 8 a ~ 8 c のうちのいずれかを指定する動的表示処理識別情報とこの動的表示処理識別情報で指定されている動的表示処理に提供される表示属性情報とを取得させる動的表示情報取得機能 3 と、動的表示情報取得機能 3 によって取得された動的表示処理識別情報で指定されている動的表示処理に対して該当する表示属性情報を提供する属性情報提供機能 4 a とを実現させる画面制御プログラム 1 を記録したコンピュータ読み取り可能な記録媒体を提供する。

【選択図】 図 1

出願人履歴情報

識別番号 [000003078]

1. 変更年月日 1990年 8月22日

[変更理由] 新規登録

住 所 神奈川県川崎市幸区堀川町72番地
氏 名 株式会社東芝